

BMSD 2016

Sixth International Symposium on Business Modeling and Software Design



Proceedings

Rhodes, Greece • 20-22 June 2016

Organized by:

Co-Organized by:

Technical
Co-Sponsorship by:

Cooperating Organizations:



BMSD 2016

Proceedings of the
Sixth International Symposium on
Business Modeling and Software Design

Rhodes, Greece

20-22 June 2016

Organized by
**IICREST - Interdisciplinary Institute for Collaboration and Research on
Enterprise Systems and Technology**

Co-Organized by
AUTH - Aristotle University of Thessaloniki

Cooperating Organizations:
TU DELFT - Delft University of Technology
**CTIT - the Center for Telematics and Information Technology of
University of Twente**
**IMI - the Institute of Mathematics and Informatics of
Bulgarian Academy of Sciences**
SIKS - the Dutch Research School for Information and Knowledge Systems
AMAKOTA Ltd.

Technical Co-Sponsorship by
BPM-D

Copyright © 2016 SCITEPRESS - Science and Technology Publications
All rights reserved

Edited by Boris Shishkov

Graphics Production by Bozhana Yankova

Compiled in Portugal

Printed in Bulgaria

ISBN: 978-989-758-190-8

Depósito Legal: 408160/16

<http://www.is-bmsd.org>

secretariat@iicrest.org

BRIEF CONTENTS

KEYNOTE SPEAKERS	IV
CHAIR AND PROGRAM COMMITTEE.....	V
BEST PAPERS SELECTION	VIII
FOREWORD.....	IX
CONTENTS.....	XIII

KEYNOTE SPEAKERS

Paris Avgeriou

University of Groningen

The Netherlands

Jan Jürjens

University of Koblenz-Landau / Fraunhofer ISST

Germany

Mathias Kirchmer

University of Pennsylvania / BPM-D

USA

CHAIR AND PROGRAM COMMITTEE

CHAIR

Boris Shishkov, Bulgarian Academy of Sciences / IICREST, Bulgaria

PROGRAM COMMITTEE

Hamideh Afsarmanesh, University of Amsterdam,
The Netherlands

Marco Aiello, University of Groningen, The
Netherlands

Mehmet Aksit, University of Twente, The
Netherlands

Antonia Albani, University of St. Gallen,
Switzerland

Paulo Anita, Delft University of Technology, The
Netherlands

Danilo Ardagna, Politecnico di Milano, Italy

Rumen Arnaudov, Technical University of Sofia,
Bulgaria

Colin Atkinson, University of Mannheim, Germany

Paris Avgeriou, University of Groningen, The
Netherlands

Csaba Boer, TBA, The Netherlands

Boyan Bontchev, Sofia University St. Kliment
Ohridski, Bulgaria

Frances Brazier, Delft University of Technology,
The Netherlands

Barrett Bryant, University of North Texas, USA

Cinzia Cappiello, Politecnico di Milano, Italy

Jorge Cardoso, University of Coimbra, Portugal

Kuo-Ming Chao, Coventry University, UK

Ruzanna Chitchyan, University of Leicester, UK

Samuel Chong, Capgemini, UK

Dimitar Christozov, American University in
Bulgaria - Blagoevgrad, Bulgaria

José Cordeiro, Polytechnic Institute of Setúbal,
Portugal

Dumitru Dan Burdescu, University of Craiova,
Romania

Jan L. G. Dietz, Delft University of Technology,
The Netherlands

Teduh Dirgahayu, Universitas Islam Indonesia,
Indonesia

Lyubka Doukovska, Bulgarian Academy of
Sciences, Bulgaria

John Edwards, Aston University, UK

Chiara Francalanci, Politecnico di Milano, Italy

PROGRAM COMMITTEE (CONT.)

Boris Fritscher, University of Applied Sciences
Western Switzerland, Switzerland

J. Paul Gibson, T&MSP - Telecom & Management
SudParis, France

Arash Golnam, Business School Lausanne,
Switzerland

Rafael Gonzalez, Javeriana University, Colombia

Clever Ricardo Guareis de Farias, University of São
Paulo, Brazil

Jens Gulden, University of Duisburg-Essen, Germany

Markus Helfert, Dublin City University, Ireland

Philip Huysmans, University of Antwerp, Belgium

Ilian Ilkov, IBM, The Netherlands

Ivan Ivanov, SUNY Empire State College, USA

Marijn Janssen, Delft University of Technology, The
Netherlands

Dmitry Kan, AlphaSense Inc., Finland

Dimitris Karagiannis, University of Vienna, Austria

Mathias Kirchmer, University of Pennsylvania /
BPM-D, USA

Marite Kirikova, Riga Technical University, Latvia

José Paulo Leal, University of Porto, Portugal

Kecheng Liu, University of Reading, UK

Leszek Maciaszek, Wroclaw University of
Economics, Poland / Macquarie University, Australia

Jelena Marincic, ASML, The Netherlands

Michele Missikoff, Institute for Systems Analysis and
Computer Science, Italy

Dimitris Mitrakos, Aristotle University of
Thessaloniki, Greece

Ricardo Neisse, European Commission Joint Research
Center, Italy

Bart Nieuwenhuis, University of Twente, The
Netherlands

Selmin Nurcan, University Paris 1 Pantheon
Sorbonne, France

Olga Ormandjieva, Concordia University, Canada

Mike Papazoglou, Tilburg University, The
Netherlands

Marcin Paprzycki, Polish Academy of Sciences,
Poland

Oscar Pastor, Universidad Politécnica de Valencia,
Spain

Barbara Pernici, Politecnico di Milano, Italy

Doncho Petkov, Eastern Connecticut State University,
USA

Plamen Petkov, Dublin City University, Ireland

Henderik Proper, Luxembourg Institute of Science
and Technology, Grand Duchy of Luxembourg

Ricardo Queirós, IPP, Portugal

PROGRAM COMMITTEE (CONT.)

Jolita Ralyte, University of Geneva, Switzerland

Gil Regev, EPFL / Itecor, Switzerland

Wenge Rong, Beihang University, China

Ella Roubtsova, Open University, The Netherlands

Irina Rychkova, University Paris 1 Pantheon
Sorbonne, France

Shazia Sadiq, University of Queensland, Australia

Valery Sokolov, Yaroslavl State University, Russia

Richard Starmans, Utrecht University, The
Netherlands

Cosmin Stoica Spahiu, University of Craiova,
Romania

Coen Suurmond, RBK Group, The Netherlands

Bedir Tekinerdogan, Wageningen University, The
Netherlands

Yasar Tonta, Hacettepe University, Turkey

Roumiana Tsankova, Technical University of Sofia,
Bulgaria

Marten van Sinderen, University of Twente, The
Netherlands

Ramayah Thurasamy, Universiti Sains Malaysia,
Malaysia

Maria Virvou, University of Piraeus, Greece

Roel Wieringa, University of Twente, The
Netherlands

Fons Wijnhoven, University of Twente, The
Netherlands

Shin-Jer Yang, Soochow University, Taiwan

Benjamin Yen, University of Hong Kong, China

Fani Zlatarova, Elizabethtown College, USA

BEST PAPERS SELECTION

The authors of around fifteen selected papers presented at BMSD 2016 will be invited by Springer-Verlag to submit revised and extended versions of their papers for publication in a Springer LNBIP Series book.

FOREWORD

This book contains the proceedings of the *Sixth International Symposium on Business Modeling and Software Design (BMSD 2016)*, held on 20-22 June 2016 in Rhodes, Greece. BMSD is an annual international scientific event that brings together researchers and practitioners who recognize as important the challenge of bringing together business/enterprise modeling and software specification. We observe that most software systems being developed need to be adequately integrated in their *enterprise context*. We observe as well that often an "enterprise context" includes among other things also already running software applications. For this reason, it is essential to have *alignment* and *traceability* between "enterprise level" and "software level", and therefore it seems logical to try identifying ENTERPRISE SYSTEMS and SOFTWARE SYSTEMS, and bridging the two on that basis. As it is well-known, when speaking of a system, we are interested in what are the *system components*, how are they *related to each other*, how are they related to the *environment*, and what are the principles guiding the *system evolution*. Here comes the "architecture" concept - an *architecture* is supposed to provide an integrated view of the system under consideration. Following one of the BMSD Keynote Lecturers, *Roel Wieringa*: an "enterprise architecture" is a *coherent whole of principles, methods and models that are used in the design and realization of the enterprise's structure, processes, (possibly) information systems, and infrastructure*. Even though *structure, processes, and data* are essential for "software architectures" as well, more complexities occur when developing software (coming through analysis, design, and implementation) - according to another BMSD Keynote Lecturer, *Leszek Maciaszek*: what lags behind is *managing system complexity expressed in terms of dependencies between system elements*. Further, current enterprises and software applications both need to be ADAPTABLE because of the constantly changing real-life environment to which they should conform. Finally, the *data aspects*, with regard to the development of information systems (as mentioned above), are an issue of increasing popularity. This is not surprising because the DATA concept relates to crosscutting concerns such as *versioning, privacy, and reliability*, and this has to be especially taken into consideration not only when modeling enterprises but also when designing software. Moreover, an adaptable information system assumes behavior adaptation based on changes in the surrounding (context) information. Among the sources of such context information are sensors – currently, there are so many sensors around resulting in data "showering"; we need to know what to do about it both at enterprise level and at software level. This all brings forward a number of BMSD-relevant *challenges*, the most important of which are:

- (i) identifying the enterprise and/or software system(s) to be considered;
- (ii) building aspect models, accordingly, including models that reflect structure, processes, data, and so on, and establishing inter-model consistency;
- (iii) capturing the (possibly different) levels of granularity between enterprise models and corresponding software models;
- (iv) establishing alignment and traceability between enterprise models and corresponding software models;
- (v) addressing possible dependencies between system elements;
- (vi) modeling adaptability;
- (vii) modeling data aspects, capturing issues, such as versioning, privacy, reliability, and so on.

Those challenges inspire the BMSD Community and are in line with the main BMSD Areas: business models and requirements, business models and services, business models and software, and information systems architectures. The last of the mentioned challenges relates to the theme of BMSD 2016:

Integrating Data Analytics in Enterprise Modeling and Software Development

In considering the above-mentioned areas and theme, it is important to stress the interdisciplinary drive of BMSD – *business modeling* and *software design*. We consider business models from the perspective of their restricting software specification and we consider software design as (ideally) stemming from underlying business models. Business models are not only about reflecting entities, processes, and rules at enterprise level but also about capturing human feelings, beliefs, intentions, and commitments. As for software specifications, they are based on algorithms and measurable inputs, and assume technical complexity, and at the same time – software specification is to be based on underlying business/enterprise models. As mentioned above, such a “bridging” between the enterprise engineering area and the information systems development area is dominant for the researchers and practitioners belonging to the BMSD Community, with BMSD demonstrating for a sixth consecutive year a high quality of papers. As always, our symposium is characterized by a stimulating discussion environment and unique networking possibilities.

The BMSD’16 proceedings consists of 27 high-quality research and experience papers that have not been published previously. Those papers have undergone a detailed peer-review process and were selected based on rigorous quality standards.

BMSD 2016 was organized and sponsored by the *Interdisciplinary Institute for Collaboration and Research on Enterprise Systems and Technology* (IICREST), being co-organized by *Aristotle University of Thessaloniki* (AUTH) and technically co-sponsored by *BPM-D*. Cooperating organizations were *Delft University of Technology* (TU Delft), the *UTwente Center for Telematics and Information Technology* (CTIT), the *BAS Institute of Mathematics and Informatics* (IMI), the *Dutch Research School for Information and Knowledge Systems* (SIKS), and *AMAKOTA Ltd.*

Since 2011, we have enjoyed five successful BMSD editions, namely: *Sofia 2011*, *Geneva 2012*, *Noordwijkerhout 2013*, *Luxembourg 2014*, and *Milan 2015*. The Rhodes 2016 edition is the sixth one and we are proud to have succeeded in establishing and maintaining high scientific quality and stimulating collaborative atmosphere. Our community is of high competence and also inspired to share ideas and experience. In addressing the above-mentioned challenges and in line with the BMSD areas, BMSD 2016 is addressing a large number of research topics: from more *conceptual ones*, such as conceptual business / enterprise modeling and meta-modeling, modeling abstractions and model transformations, modeling notations, inter-model consistency, knowledge modeling, context-aware systems, sign systems and value modeling, capability-affordance modeling, and

cost modeling, to more *technical ones*, such as requirements identification and software specification, (model-driven) software development, software product line engineering, software re-use and design patterns detection, applications routing, information security and security certification, technical debt, SaaS, e-Business and service e-Marketplaces, and socio-technical information systems, from topics *exclusively focusing on Business Processes* (BP), such as BP modeling, BP modeling formalisms and BP modeling tools, BP mining and BP management, declarative business rules, BP inconsistency risk detection, sensitive BPs, to topics related to *architectures*, such as enterprise architectures and visualization of changes in enterprise architectures, service-oriented architectures and service orchestration addressing also micro-services, with all those topics reflected in a number of application domains; among those *domains* are: ambient assisted living, border security, consultancy, healthcare, retail, 3D printing, technology-enhanced learning, and voting. Finally, the special session on *Green IT Solutions* is also appreciated in this regard, featuring topics, such as network performance modeling (reflecting energy consumption) and (e-car) routing modeling. We bring forward compliments to authors for their inspiring contributions, proposing ideas of high innovative potential.

BMSD 2016 received 59 paper submissions from which 27 papers were selected (including several invited papers) for publication in the current proceedings. 17 of those papers were selected for a 30-minute oral presentation (full papers), leading to a full-paper acceptance ratio of 29% (compared to 24% in 2015). In addition: 6 papers were selected for a 20-minute oral presentation (short papers and special session papers); 4 papers were selected for a poster presentation. The above-mentioned full-paper acceptance ratio shows a high BMSD quality which we intend to maintain and reinforce in the following editions of the symposium. Further, the BMSD'16 authors / keynote lecturers are from: Austria, Belgium, Bulgaria, Colombia, Greece, Finland, Germany, Japan, The Netherlands, Poland, Portugal, Sweden, Switzerland, Tunisia, UK, and USA (listed alphabetically); that makes a total of 16 countries (compared to 21 in 2015, 21 in 2014, 14 in 2013, 11 in 2012, and 10 in 2011) to justify a strong international presence. Finally, 6 countries have been represented at all 6 BMSD editions so far – those are: *Belgium, Bulgaria, Germany, Switzerland, The Netherlands*, and *UK*, to indicate a strong European influence.

Publisher of the current proceedings is SCITEPRESS. Besides printed proceedings, we also deliver an electronic version – all presented papers will be made available at the *SCITEPRESS Digital Library* by September, 2016. Furthermore, the proceedings will be submitted to *DBLP* (Computer Science Bibliography) for indexation. Finally, the authors of around fifteen selected papers presented at BMSD 2016 will be invited by *Springer-Verlag* to submit revised and extended versions of their papers for publication in a Springer LNBIP (Lecture Notes in Business Information Processing) Series book.

The high quality of the BMSD 2016 program is enhanced by three keynote lectures, delivered by distinguished guests who are renowned experts in their fields: *Paris Avgeriou* (University of Groningen), *Jan Jürjens* (University of Koblenz-Landau / Fraunhofer ISST), and *Mathias Kirchmer* (University of Pennsylvania / BPM-D). The keynote speakers and some other BMSD'16 participants will take part in a panel discussion and also in other discussions stimulating community building and facilitating possible R&D project acquisition initiatives. Those special activities will definitely contribute to maintaining the event's high quality and inspiring our steady and motivated community.

Organizing this interesting and successful symposium required the dedicated efforts of many people. Firstly, we must thank the authors, whose research and development achievements are recorded here. Next, the program committee members each deserve credit for the diligent and rigorous peer-reviewing. Further, we would like to mention the excellent organization provided by the *IICREST* team (supported by its logistics partner, *AMAKOTA Ltd.*) – the team did all the necessary work for delivering a stimulating and productive event, and we have to acknowledge also the support of *Dimitris Mitrakos* and *Vassilis Pappas*. We appreciate the cooperation of our colleagues from *Aristotle University of Thessaloniki* and *Delft University of Technology*. We are grateful to *SCITEPRESS* for their willingness to publish the current proceedings and we bring forward special compliments to *Vitor Pedrosa* for his professionalism, supportiveness, and excellent collaboration with regard to the proceedings preparation. Last but not least, we thank the keynote speakers for their invaluable contribution and for taking the time to synthesize and deliver their talks.

We wish you all an inspiring symposium and an enjoyable stay in beautiful Rhodes. We look forward to seeing you next year in Barcelona, Spain, for the Seventh International Symposium on Business Modeling and Software Design (BMSD 2017), details of which will be made available on <http://www.is-bmsd.org>.

Boris Shishkov

Bulgarian Academy of Sciences / IICREST, Bulgaria

CONTENTS

KEYNOTE SPEAKERS

Technical Debt - How Software Organizations Can Stay Solvent <i>Paris Angerou</i>	3
15 Years of Model-Based Security Engineering with UML - Supporting Secure Evolution <i>Jan Jürjens</i>	5
The BPM-Discipline - Strategy Execution in a Digital World <i>Mathias Kirchmer</i>	7

FULL PAPERS

Interactions, Transitions and Inference Rules in Semantically Integrated Conceptual Modelling <i>Remigijus Gustas and Prima Gustienė</i>	11
Modelling Information Systems using NOMIS - An Overview of its Modelling Notation and Implementation <i>José Cordeiro</i>	22
Hard Problems and Soft Information <i>Coen Suurmond</i>	32
Assessing Business Processes by Checking Transaction Documents for Inconsistency Risks <i>Takafumi Komoto, Kokichi Futatsugi and Nobukazu Yoshioka</i>	39
A Metamodel for Business Rules with Access Control <i>Lex Wedemeijer</i>	46
Business Models & Business Cases for Point-Of-Care Testing <i>A. J. Starling, L. O. Meertens and N. Sikkkel</i>	54
On the Effect of Digital Frontstores on Transforming Business Models - Concept and Use-case from the Consulting Industry <i>Denis Jobann, Tobias Greff and Dirk Werth</i>	64
3D Printing and Additive Manufacturing Capability Modelling <i>Vaughan Michell</i>	73
A Multi-criteria Evaluation Framework for Selecting Sensitive Business Processes Modeling Formalism <i>Mariam Ben Hassen, Mohamed Turki and Faïez Gargouri</i>	84
A Proposal to Model Knowledge in Knowledge-Intensive Business Processes <i>Norbert Gronau, Christoph Thim, André Ulrich, Gergana Vladova and Edzard Weber</i>	98
Characteristics of Enterprise Architecture Analyses <i>Julia Rauscher, Melanie Langermeier and Bernhard Bauer</i>	104
Visualization of Changes in ArchiMate - Within the Application and Technology Layers <i>Robert Bakelaar and Ella Roubtsova</i>	114
An in Depth Comparative Analysis of Software Tools for Modelling and Simulating Business Processes <i>Rallis C. Papademetriou and Dimitrios A. Karras</i>	124

Microflows: Lightweight Automated Planning and Enactment of Workflows Comprising Semantically- Annotated Microservices <i>Roy Oberhauser</i>	134
Scheduling Different Types of Applications in a SaaS Cloud <i>Georgios L. Stavrinides and Helen D. Karatza</i>	144
On Service e-Marketplaces for Independent and Assisted Living <i>Leszek A. Maciaszek</i>	152
DP-CORE: A Design Pattern Detection Tool for Code Reuse <i>Themistoklis Diamantopoulos, Antonis Noutsos and Andreas Symeonidis</i>	160
SHORT PAPERS	
Rule-based Business Process Abstraction Framework <i>Christina Tsagakani and Aphrodite Tsalgatidou</i>	173
Cross-System Process Mining using RFID Technology <i>Christian Glaschke, Norbert Gronau and Benedict Bender</i>	179
Towards a Service-Oriented Architecture for eVoting <i>Boris Shishkov and Marijn Janssen</i>	187
A Framework for Process Driven Software Configuration <i>Andreas Daniel Sinnhofer, Peter Pühringer, Klaus Potzmader, Clemens Orthacker, Christian Steger and Christian Kreiner</i>	196
POSTERS	
Towards Context-aware Border Security Control <i>Boris Shishkov and Dimitris Mitrakos</i>	207
Introduction of BEMES, a Webtool to Simplify Business Process and Equipment Cost Modelling <i>Jonathan Spruytte, Marlies Van der Wee, Sofie Verbrugge and Didier Colle</i>	213
Using a Multi-Perspectives Approach for Building a Socio-Technical Information System <i>Sayeb Belbassen Yemna, Foughali Ibtissem and Hajjami Ben Ghezala Henda</i>	217
Metaprocesses as Software Assets Reuse <i>Javier Darío Fernández-Ledesma</i>	221
SPECIAL SESSION ON GREEN IT SOLUTIONS	
The Effect of Network Performance on High Energy Physics Computing <i>Jukka Kommeri, Aleksi Vartiainen, Seppo Heikkilä and Tapio Niemi</i>	227
A Backpressure Framework Applied to Road Traffic Routing for Electric Vehicles <i>Evangelos D. Spyrou and Dimitrios K. Mitrakos</i>	235
AUTHOR INDEX	241

KEYNOTE SPEAKERS

Technical Debt

How Software Organizations Can Stay Solvent

Paris Avgeriou
University of Groningen, The Netherlands
paris@cs.rug.nl

Abstract: The term *Technical Debt* has become rather popular over the past years, expressing technical compromises that can yield short-term benefits but may hurt the long-term health of a software system. There are good news: Technical Debt as a metaphor resonates well with technical and non-technical stakeholders, and can potentially act as a bridge between them and facilitate communication and negotiation. There are also bad news: Technical Debt is undeniably accumulating in most large systems, pervading the entire lifecycle from requirements to deployment; it threatens to “bankrupt” those systems if it is not actively managed. The future of software engineering research and practice will revolve around how to identify, measure, prioritize and repay Technical Debt, as well as how to make sound investments to balance short- and long- term goals. In this talk, we revisit the state-of-the art and practice to examine how much progress is achieved so far, and we discuss some promising future directions in the field, concluding with a “call to arms”.

BRIEF BIOGRAPHY

Dr. Paris Avgeriou is Professor of Software Engineering in the University of Groningen, the Netherlands where he has led the Software Engineering research group since September 2006. Before joining Groningen, he was a post-doctoral Fellow of the European Research Consortium for Informatics and Mathematics (ERCIM). He sits on the editorial board of IEEE Software and Springer Transactions on Pattern Languages of Programming. His research interests lie in the area of software architecture, with strong emphasis on architecture modeling, knowledge, evolution, patterns and link to requirements. He champions the evidence-based paradigm in Software Engineering research.

15 Years of Model-Based Security Engineering with UML

Supporting Secure Evolution

Jan Jürjens

University of Koblenz-Landau / Fraunhofer ISST, Germany
juerjens <at> uni <dash> koblenz <dot> de

Abstract: Security certification of complex systems requires a high amount of effort. As a particular challenge, today's systems are increasingly long-living and subject to continuous change. After each change of some part of the system, the whole system needs to be re-certified from scratch (since security properties are not in general modular), which is usually far too much effort. There has been recent work to address this challenge in the context of a line of work which develops approaches and tools for Model-based Security Engineering, making use of established modeling notations such as the Unified Modeling Language (UML). From that work, this talk presents a tool-supported approach for security certification that minimizes the amount of effort necessary in the case of re-certification after change. It is based on results that determine under which conditions change preserves security properties (for example in the context of structuring techniques such as refinement or architectural principles such as modularization). The approach supports an automated difference-based security analysis, at the level of design models as well as the implementation code (using static security analysis or run-time verification). It has been applied e.g. to cryptographic protocols, distributed security infrastructures, and identity management systems, and there are empirical results comparing it to classical techniques for security certification. In the outlook, we briefly present current research directions, such as applying the approach to the security certification of the Industrial Data Space (currently in development by Fraunhofer and a consortium of more than 20 companies, see <http://www.industrialdataspace.org/en>).

BRIEF BIOGRAPHY

Jan Jürjens is Professor of Software Engineering at the University of Koblenz-Landau (Germany) and Director Research Projects at Fraunhofer Institute for Software and Systems Engineering ISST (Germany). He has been PI of various projects, often in cooperation with industry (e.g. Microsoft Research (Cambridge)). Previous positions include Professor at TU Dortmund, a Royal Society Industrial Fellowship at Microsoft Research Cambridge, a non-stipendiary Research Fellowship at Robinson College (Univ. Cambridge), where in 2009 he was appointed as Senior Member, and a Postdoc position at TU Munich. Jan holds a Doctor of Philosophy in Computing from University of Oxford and is author of "Secure Systems Development with UML" (Springer, 2005; Chinese translation 2009) and other publications mostly on software engineering and IT security. More information: <http://jan.jurjens.de>.

The BPM-Discipline

Strategy Execution in a Digital World

Mathias Kirchmer

University of Pennsylvania / BPM-D, USA

Mathias.Kirchmer@bpm-d.com

Abstract: According to a recent study of The Gartner Group only 13% of business meet their yearly strategic goals. This means 87% of organizations prepare strategic plans and related goals – but they don't meet their goals. In addition, less than 1% of companies have prepared their business processes to realize the potential of our digital world. Hence the risk of not executing successfully on a business strategy becomes even higher. This presentation will show how the discipline of business process management (BPM) addresses those issues. It helps organizations targeting value in a digital world through cross-functional integrated business and technology initiatives. BPM becomes the management discipline of strategy execution. The presentation discusses the value and key components of the discipline of process management. It introduces a patent-pending framework to establish and manage a BPM-Discipline successfully. This management discipline helps companies to organize their “process of process management” resulting in increased productivity and performance – and at the end in a systematic execution of their strategy. We will discuss the research prototype of a next generation software application supporting the application of the framework. The BPM-Discipline leverages the opportunities of the digital world systematically to deliver business value fast and at low risk, by executing strategy systematically.

BRIEF BIOGRAPHY

As innovative and pragmatic top executive, including roles as CEO and Managing Director, Dr. Kirchmer has worked successfully in an international environment. He is a visionary leader, thought leader and innovator in the field of Business Process Management (BPM). Dr. Kirchmer has combined his broad practical business experience with his extensive academic research, leading to pioneering management approaches that have proven to be both, sustainable and provide immediate benefits. His deep and layered knowledge of BPM has proven successful with small and large companies in various industries around the world, including Germany, France, USA, Brazil, Chile, Japan, and India. Most recently, Dr. Kirchmer has founded BPM-D, a company focused on enabling the next generation enterprise by leveraging the discipline of BPM. Before Dr. Kirchmer has been Accenture's Managing Director & Global Lead for BPM. Prior to joining Accenture, Dr. Kirchmer was the CEO of the Americas & Japan and The Chief Innovation & Marketing Officer for IDS Scheer, a leading provider of software and consulting

solutions for BPM, best known for its ARIS software. Dr. Kirchmer remains involved in academia as an affiliated faculty member at the University of Pennsylvania since 1998, the Business School of Widener University, Philadelphia University and the Universidad of Chile as a visiting professor. In 1984, he received a research and teaching fellowship from the Japan Society for the Promotion of Science. Dr. Kirchmer is a published authority of BPM authoring 6 books as well as over 120 articles and public presentations. Dr. Kirchmer holds a PHD in Information Systems from Saarbrücken University, a Master in Business Informatics from Karlsruhe Technical University, as well as a Master in Economics from Paris-IX-Dauphine University. He resides in West Chester, Pennsylvania and has a US and German citizenship.

FULL PAPERS

Interactions, Transitions and Inference Rules in Semantically Integrated Conceptual Modelling

Remigijus Gustas and Prima Gustiené

Department of Information Systems, Karlstad University, Sweden
{Remigijus.Gustas, Prima.Gustiene}@kau.se

Keywords: Service Interactions, Value Exchanges, Inference Rules, Object Transitions, Integration of Static and Dynamic Aspects.

Abstract: To obtain value from the graphical representations that are used by different stakeholders during the system development process, they must be integrated. This is important to achieve a holistic understanding about system specification. Integration can be reached via modelling process. Currently, most of information system modelling methods present different modelling aspects in disparate modelling dimensions and therefore it is difficult to achieve semantic integrity of various diagrams. In this paper, we present semantically integrated conceptual modelling method for information system analysis and design. The foundation of this modelling method is based on interactions. This way of modelling provides possibility of integration of business processes and business data. The inference rules of interactions help in reasoning about the decomposition of concepts. In this method, decomposition of the system is graphically described as classification, inheritance or composition of organizational and technical system components.

1 INTRODUCTION

Conceptual modelling is a fundamental activity in requirements engineering (Nuseibeh and Easterbrook, 2000). It is the act of abstracting a model from a problem domain (Lankhorst, 2005). One of the main problems in conceptual modelling of Information Systems (IS) is that conventional modelling methods define different aspects of a system using different types of diagrams. Integration principles of such diagrams are not clear. The lack of a conceptual modelling method that helps to detect semantic integrity of IS specifications is a big information systems development problem. Semantically Integrated Conceptual Modelling (SICM) method challenges the existing integration problems among interactive, behavioural and structural aspects (Gustas and Gustiene, 2012) of IS. To capture the holistic structure of a system, it is necessary to understand how various components are related.

To obtain value from graphical representations that are used in an organisation by different stakeholders, these representations must be integrated. Integrated enterprise models might help business and information technology experts to communicate in order to assess and trace the impact

of organizational changes. Integration can be reached via modelling process. Modelling helps system developers to visualize, specify, construct and document different aspects of the system. Modelling is the only way to control system development process. Various aspects of the system may have many modelling projections, which are typically described by using different types of diagrams. These diagrams are critical to distinguish between disparate dimensions of enterprise architecture (Zachman, 1987). The Zachman Framework (1987) can be viewed as taxonomy for understanding different types of diagrams. This framework defines separate dimensions of business application and data architecture, such as Why, What, How, Who, Where and When. Inability to detect inconsistency among different architecture views and dimensions is one of the fundamental problems in information system methodologies.

Most conventional conceptual modelling languages are plagued by the semantic mismatch between static and dynamic constructs of meta-models. To achieve semantic integration in such a case is very difficult. Unified Modelling Language (UML) (OMG, 2009) uses various types of diagrams to represent behavioural, structural and interaction aspects of the system. Every modelling approach

that covers more than one type of requirements and is represented by the collection of different diagrams must contain the systematic method for the detection of inter-model inconsistency. The static aspects describe characteristics of objects, which are invariant in time. The dynamic aspects describe interactive and behavioural characteristics of objects over time. These aspects are complimentary and they cannot be analysed in isolation.

Inter-model consistency and completeness of system specifications is hard to achieve for non-integrated model collections (Glinz, 2000). Modelling techniques that are realized as collection of models are difficult to comprehend for business experts. There are often semantic discontinuity and overlapping in various specifications, because static and dynamic constructs do not fit perfectly. A number of rules are defined for UML (Evermann and Wand, 2009) that are not supported by available CASE tools. Thus, working with the collections of non-integrated models causes difficulties to realize semantic quality of system specifications, which are represented on various levels of abstraction. By modelling isolated IS views and dimensions creates difficulties for business experts, who determine the organizational strategies. Consequently, this isolation increases semantic problems of communication between business experts and IT-system designers.

The SICM method provides several advantages (Gustas, 2010). Since the method is based on a single diagram type, the integrity rules can be introduced directly into one model. Particular views of specific diagram types, which define structural, behavioural or interactive aspects, can be generated by producing projections of one integrated model. In this paper, we demonstrate how the SICM method can be applied for integration of behavioural and structural aspects of conceptual representations. Given the central role of service concept in this study, it provides us with a possibility to model the most essential parts of the system, which is composed of organizational or technical components. This way of modelling is more comprehensible not just for IS designers, but also for business modelling experts, who are mostly interested in computation-neutral analysis of organizations. The presented SICM method shares many similarities with ontological foundation of service process (Ferrario and Guarino, 2008). Nevertheless, the internal behaviour of service is analysed by using the basic principles of an ontological framework, which is developed by Bunge (Bunge, 1979).

This paper is organized as follows. In the next section, some deficiencies of conceptual modelling approaches are described. How value exchanges are decomposed into different parts is discussed in the third section. In the fourth section, various types of conceptual dependencies and their inference rules are described. And finally, we present the conclusions of this work.

2 DEFICIENCIES OF CONCEPTUAL MODELLING APPROACHES

Conceptual modelling still lacks the methods that provide a possibility to model different problem domains in an integrated way. Integrated graphical representation of business process and business data is very relevant for reasoning about enterprise redesign decisions. As all steps in SICM method uses the same model and the same way of modelling that is based on service interaction flows. It enables enterprise architects to gradually decompose a system and to move smoothly from system analysis to design without being required to represent a complete solution. UML (OMG, 2009) was developed with the ultimate goal to unify the best features of the graphical modelling languages and create a de facto industry standard for system development. However, the semantic integration principles of different UML diagram types are not sufficiently clear. UML models have several weaknesses, which can be summarized as follows: value flow exchanges between actors cannot be explicitly captured; system decomposition principles are ambiguous; it is unclear how to integrate interactive, structural and behavioural aspects together in a single view.

Data flow modelling and clear system decomposition principles were applied in structured analysis and design methods (Gane and Sarson, 1979). UML also supports various types of associations between classes, actors, or between software or hardware components. However, these methods are not suitable for modelling the direct communication among actors that define actor interactions outside the technical system boundary. It is unclear how to visualize the rich context of actor interactions, which are important components in any system. If we have no method how to explicitly capture actors and their interactions, then this important part of specification, which may be viewed as a tacit knowledge, will be hidden from enterprise architects.

One of the benefits of enterprise modelling is the ability to analyse business processes for reaching agreement among various stakeholders on how and by whom the processes are carried out. The industrial versions of information system modelling methods that are intended for business process modelling do not explicitly use the concept of value flow. Value models, which include resource exchange activities among actors, can be viewed as design guidance. The declarative nature of value flows is very useful from the system analysis point of view for the simple reason that flows have very little to do with the dependencies between business activities. Each value flow between actors, that can play the role of service requester and service provider, can be further refined in terms of more specific coordinating interactions among organizational components. The way of modelling, which is based on service flows, is more comprehensible and thus more suitable to discuss changes of process architectures with business developers, enterprise architects, system designers and users. Business process modelling does not deal with the notion of value flow, which demonstrates value exchange among actors involved (Gordijn et al., 2000). Traditionally, information system methodologies are quite weak in representing the alternative value flow exchange scenarios, which usually represent the broken commitments.

Bunge (1979) provides one of the most general ontological definitions of a system. In this paper, his definition serves as the theoretical basis for understanding the notions of organization and enterprise ontology (Dietz, 2001). Bunge's ontological principles are fundamental for the justification of various conceptual modelling constructs in our semantically integrated modelling method (Gustas and Gustiene, 2012). These principles are as follows: enterprise system can be decomposed into subsystems, which are viewed as interacting components; every subsystem can be loosely coupled with interactions to other subsystems; when subsystems interact, they cause certain things to change and changes are manifested via properties.

Any subsystem can be viewed as an object, but not every object is a subsystem. According to Bunge, only interacting objects can be viewed as subsystems. It is quite beneficial to specify service interactions and to keep track of crosscutting concerns (Jacobson and Ng, 2005) between different subsystems in order to justify their usefulness. However, a basic underlying principle in UML is to provide separate models for different aspects. It is

not totally clear how these aspects can be merged back into one model. Subsystems in UML cannot be realized as composite classes. UML does not provide any superimposition principles of static and dynamic aspects. There is very little research done on how the structural aspects and state dependent behaviour of objects should be combined with use case models. Classes and their associated state machines are regarded as the realization of use cases. Use case diagrams are typically not augmented with specification of state related behaviour (Glinz, 2000).

System decomposition should be strictly partitioned. Every component partitions a system into parts, which can be loosely coupled with other components without detailed knowledge of their internal structure. Object transitions and structural aspects have to be related to one separate service, which consists of organizational or technical components. The limitation of conventional system modelling methods results in two side effects, better known as tangling and scattering in aspect-oriented software development (Jacobson and Ng, 2005). The treatment of these deficiencies requires the modification of UML foundation. Introducing fundamental changes into UML syntax and semantics with the purpose of semantic integration of collections of models is a complex research activity. However, such attempts would allow using UML to provide computation-neutral type of diagrams, which are more suitable to reason about enterprise architectures. It is recognized that UML support for such task is vague, because semantic integration principles of different diagram types are still lacking (Harel and Rumpe, 2004).

3 VALUE FLOW EXCHANGES AND TRANSITIONS

Semantically integrated conceptual modelling paradigm is based on more rigorous interpretation of human work. A new conception helps us to develop the method of enterprise engineering that allows practitioners to see the sources of breakdowns, the connections to systems design and to guide the redesign of work processes towards greater productivity and customer satisfaction. Business process models of organizations are quite good for viewing moving material and information flows, but they provide no mechanism for ensuring that the service requester is satisfied. Service requesters deal with work processes to be done, agreements on what

will be done, who will do it, and whether they are satisfied with what has been done. The movement of information or material flows is a consequence of this work. Service flow modelling is quite intuitive way of system analysis that is easy to understand for business experts and information system designers. Actions in services are required for exchange of business flows. Actions together with exchange flows can be viewed as fundamental elements for specifying business process scenarios. A scenario is an excellent means of describing the order of service interactions. Scenarios help system designers to express business processes in interplay with elementary service interactions between enterprise system components. In such a way, value flows and service interactions provide a natural way of process decomposition.

The technologies to model coordination processes and tracking events have not been available till now. There are some concepts such as commitment and contract that are present in all business scenarios. Understanding these concepts makes it much easier to design and to change systems under construction. Commitment is a promise or obligation of an actor to perform a specific action. Contract is an agreement between service requester and service provider to exchange one asset into another. Thus, the contract may specify what happens if the commitment is not fulfilled. According to McCarthy (1982), the contract consists of increment and decrement events. If an enterprise increases one resource in an exchange, it has to decrease the value of another resource. The contract includes (1) transfer of economic resources, (2) transfer of exchange rights. Any exchange is a process, in which an enterprise receives economic resource and in return gives other resources. For example, a contract contains commitments to sell goods and to receive payments. The terms of the sales order can specify penalties if goods or payments have not been received on time. The creation and termination of primary business data in these exchanges are important for an enterprise. Artefacts such as credit, debit, account balances are derived from these exchanges.

Interaction dependencies are important to conceptualize business processes as services between various enterprise actors. Since actors can be implemented as organizational or technical system components, these components can interact according to the prescribed service interaction patterns to achieve their goals. In SICM, the general service interaction pattern is represented by two interaction dependencies into opposite directions

between two actors: service requester and service provider (Gustas, 2010). The idea of this pattern is similar to a well-known DEMO transaction pattern (Dietz, 2006). The SICM pattern is illustrated graphically in figure 1.

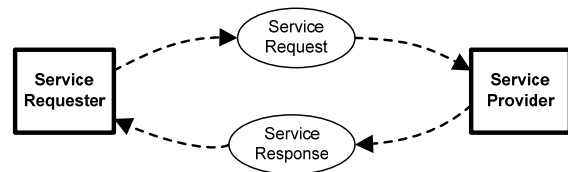


Figure 1: Elementary service interaction loop.

Interaction loop between two actors indicates that they depend on each other by specific actions. Service providers are actors who typically receive service requests, over which they have no direct control. They initiate service responses that are sent to service requesters. These two interacting actors can be used to define more complex interaction activities. Using interaction pattern, as way of modelling, enables system designers to construct the blueprint of interacting components, which can be represented by different actors across organizational and technical system boundaries. Any enterprise system can be defined as a set of interacting and loosely connected components, which are able to perform specific services on request.

Increment and decrement events represent values exchanged in business processes. Value models (Gordijn et al., 2000) clarify why actors are willing to exchange economic resources with each other. Actors, actions and exchange flows are elements that are necessary for demonstrating value exchange. Economic resources are special types of concepts, which represent moving things. Rectangles, with shaded background, are used to represent economic resources and dotted line boxes are used for the representation of exchange flows. Actors are represented by square rectangles and actions are represented by ellipses. Actions that are performed by actors are necessary for transferring economic resources, data or decision flows between actors. Two actors and transfer of value flows into opposite directions is illustrated in figure 2.

This figure illustrates that Deliver and Pay actions may happen at any time. It is not stated, which action should happen first. We just want to show that a customer is exchanging a Payment flow into a Delivery flow. Deliver action is initiated by vendor, because a shipment's moving direction is from Vendor to Customer. On the contrary, the payment is moving from customer to vendor through

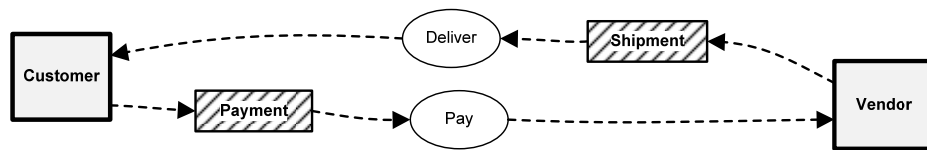


Figure 2: Value exchange.

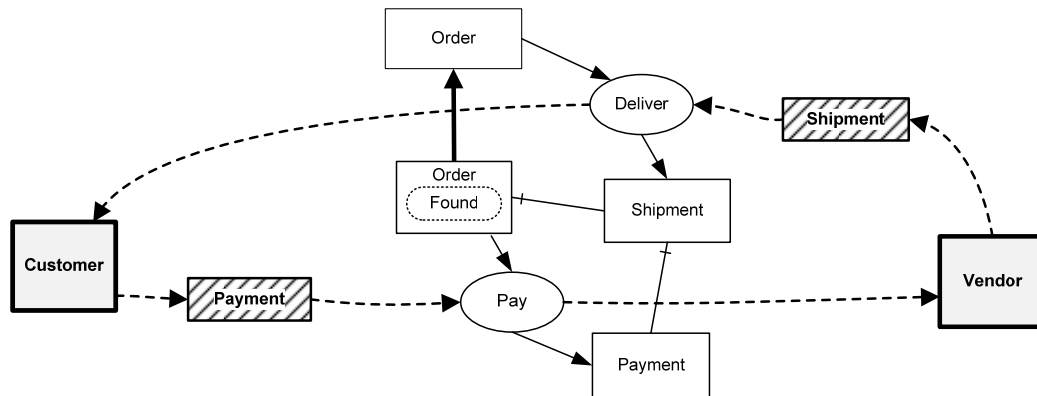


Figure 3: Interaction loop, where Deliver action precedes Pay action.

the action of pay. Action of Pay and Action of Deliver represent increment and decrement events. The process of paying is essentially the exchange of Shipment for Payment from the point of view of both actors. For a Vendor, the pay action is an increment event and deliver action is a decrement event, because it decreases the value of resources under control. For a Customer, it is vice versa. The terms of increment and decrement actions depend on the actor, which is the focus of this model.

Our buying and selling example focuses on the core phenomenon. Most customers pay in advance for shipment, but some customers want to pay, just when they receive the delivered products. If we consider the case of online sales, then customers provide credit card details before the product items are delivered. Some customers receive an invoice later and pay for all their purchases in a certain period. All these cases are covered by the same service interaction pattern, which is illustrated in Figure 2. When shipment is delivered, then the delivery fact is registered in a system by a newly created object with its mandatory properties. The transition arrow (\rightarrow) is pointing to the class, which represents the creation of a new object. In our example, it is an object of Shipment. The fact of money transfer from Customer to Vendor is represented by Payment. If we want to represent that Deliver action precedes Pay Action, then the created Shipment itself or some of its property should be linked by the transition arrow with the Pay Action, which indicates the creation of the next object. The

pay action creates the Payment from the Order [Found] object, which is the property of Shipment. It is represented in figure 3.

We may want to ask for payment in advance of shipment. In this case, we show the first action of pay, which is designed to transform the concept of invoice (not shown in our example) to payment. The second action of deliver should be connected through a transition arrow from payment to shipment. In this way, material flow of payment would be exchanged for the shipment flow.

Creation action is represented by a transition into an initial class. Termination action can be represented by transitions from a final class. If termination and creation actions are performed at the same time, then such action is called a reclassification. For instance, the initiation of the order action is typically used to create an Order record in a Vendor database. If customer Order is accepted, then it may be used for triggering the send invoice action. The internal changes are expressed by using transition links between various classes of objects in figure 4.

Creation and termination actions are used together with the object flows. In interaction pattern, a transition arrow to action or transition arrow from action represents a control flow. In such a way, any communication action can be used to superimpose interactions and control flow effects in the same diagram. Order is created by the Order Delivery action and then it is reclassified to Invoice by the send invoice action.

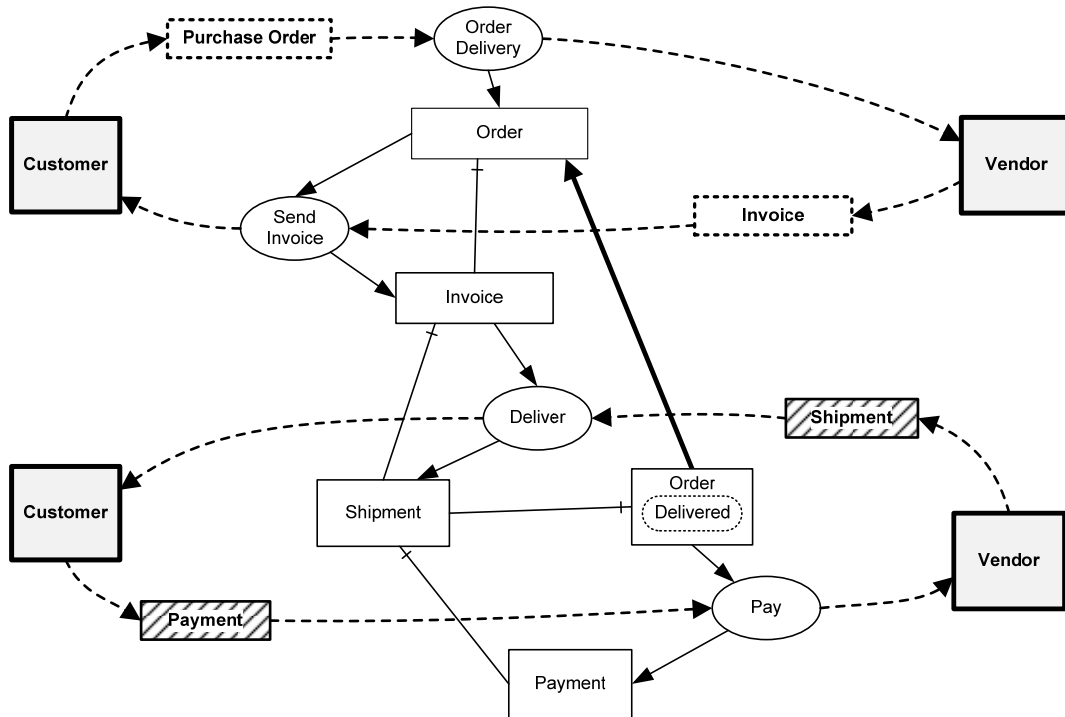


Figure 4: Example of two interaction loops with the creation and reclassification actions.

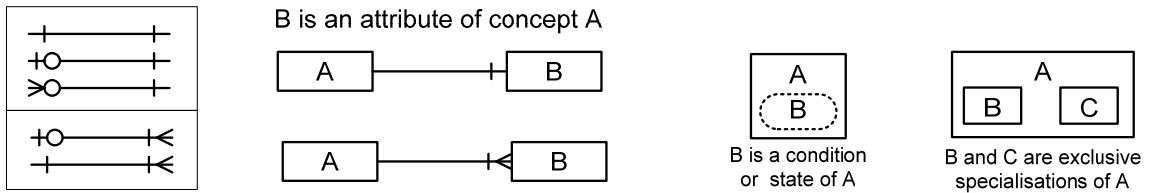


Figure 5: Notation of attribute dependencies.

The reclassification is defined as termination of object in one class and the creation of object in another class. An invoice object is created from the moving invoice flow, which represents data at rest. In the second interaction loop, a vendor delivers shipment to a customer. The delivery action corresponds to the performance act, which produces the result. It is represented by the object of shipment. Finally, the pay action indicates an acceptance of the delivered result. At the same time, it is a second performing action, which represents an exchange of shipment for payment.

Actors represent physical subsystems and structural changes of concepts represent static aspects of a system. This way of modelling allows illustrating actions, which result in changes of the attribute values. All actions are used to show the legal ways in which actors interact with each other. Structural changes of objects can be defined via static properties of objects. They are represented by

the mandatory attributes. The mandatory attributes are linked to classes through the single-valued or through multi-valued attribute dependencies. One significant difference of the presented modelling approach is that the association ends of static relations are nameless. The justification of this way of modelling can be found in (Gustas and Gustiene, 2012).

The main reason for introducing nameless attribute dependencies is to improve the stability of conceptualizations. Semantics of static dependencies are defined by cardinalities, which represent a minimum and maximum number of objects in one class (B) that can be associated with the objects in another class (A). Single-valued dependency is defined by the following cardinalities: (0,1;1,1), (0,*;1,1) and (1,1;1,1). Multi-valued dependency denotes either (0,1;1,*) or (1,1;1,*) cardinality. Graphical notation of an attribute dependency between A and B is represented in figure 5.

According to the ontological principles, which are developed by Bunge (Bunge, 1977), the structural changes of objects can be presented via object properties. Properties can be understood as mandatory attribute values. If diagrams are used to communicate unambiguously the semantic details of a conceptualized system, then optional properties should be proscribed (Gemino, 1998). If B is dependent on A, then concept A is viewed as a class and concept B is viewed as a property of A. Any concept can be defined as an exclusive complete generalization of two concepts. Concept can also be characterized by state (Dori, 2002) or condition (Gustas, 2010). Notation of exclusive generalization and notation of state are presented in figure 5 as well.

4 INFERENCE RULES OF INTERACTIONS

A model of a system can be analysed as the composition of organizational and technical components. These components represent various types of actors. Organizational components can be seen as interacting subsystems such as individuals and divisions, which denote groups of people. Technical components can be seen as interacting subsystems such as machines, software and hardware. SICM method distinguishes between two types of concepts: active and passive (Gustas, 2010). Actors can be represented just by active concepts. An instance of any actor is an autonomous subsystem. Its life cycle can only be motivated by a set of interaction dependencies with other actors. Actors are represented by non-overlapping subsystems. Classes of objects, which represent persistent data, are denoted by passive concepts. Mandatory attributes characterize all passive concepts. The objects that are represented by passive concepts can be affected by various interactions. Passive concepts can be related by the static relations such as classification, inheritance and composition.

Classification dependency ($\bullet\text{---}$) specifies objects or subsystems as instances of concepts. Classification is often referred to as instantiation, which is the reverse of classification dependency. Object-oriented approaches treat a classification relation as a more restrictive. It can only be defined between a class and an object. A class cannot play the role of object. In SICM method, each concept can be interpreted again as an instance (Gustas,

2010). For example; MS Outlook $\bullet\text{---}$ E-mail Application, E-mail Application $\bullet\text{---}$ Product Type, Product Type $\bullet\text{---}$ Concept.

Composition ($\text{---}\blacktriangleright\text{---}$) dependency in SICM method is much stronger form of aggregation, and differs significantly from the object-oriented composition. Composition dependency in SICM method allows just 1 or 1...* cardinalities between wholes and parts. This means that any part cannot be optional. The distinctive and very important features of this type of composition are as follows (Guizzardi, 2007):

- each part is existentially dependent on a whole, if a whole has a single part, then this part has coincident lifetime with a whole,
- if a whole has more than one part, then creation of a first part is coincident with the creation of a whole,
- removal or creation of additional parts can take place any time, but removal of a last part is coincident with the removal of a whole.
- creation or removal of a whole can be done together with all its parts,
- part may belong just to one and the same whole.

The definition of composition in general is not so strict. With the help of special modelling techniques, other cases of aggregation can be changed into this strict kind of composition (Gustas, 2010).

Composition hierarchies can be used for detection of inconsistent interaction dependencies between actors. Loosely coupled actors never belong to the same decomposition hierarchy. Interaction dependencies among loosely coupled actors on the lower level of decomposition are propagated into compositional wholes. So, composition links can be used for reasoning about derived interaction dependencies between actors on the higher granularity levels of specification. Interaction dependencies between actors, which are placed on two different composition hierarchies, are characterized by the following inference rules:

- if Action($X \text{---}\blacktriangleright Z$), Action($C1 \text{---}\blacktriangleright C2$) and $X \text{---}\blacktriangleright Y$
then Action($Y \text{---}\blacktriangleright Z$),
- if Action($Z \text{---}\blacktriangleright X$), Action($C2 \text{---}\blacktriangleright C3$) and $X \text{---}\blacktriangleright Y$
then Action($Z \text{---}\blacktriangleright Y$).

Interaction dependency Action($X \text{---}\blacktriangleright Z$) between two actors X and Y indicates that subsystem denoted by X is able to perform an action on one or more subsystems of Z. Action($X \text{---}\blacktriangleright Z$) represents base interaction dependency and Action($Y \text{---}\blacktriangleright Z$), which

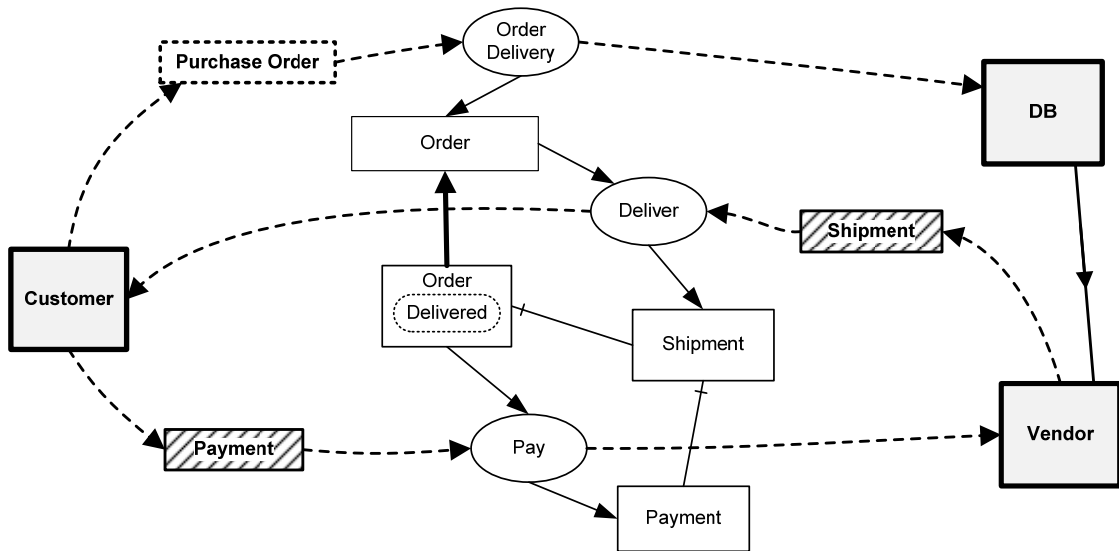


Figure 6: Base interactions between Customer and Vendor.

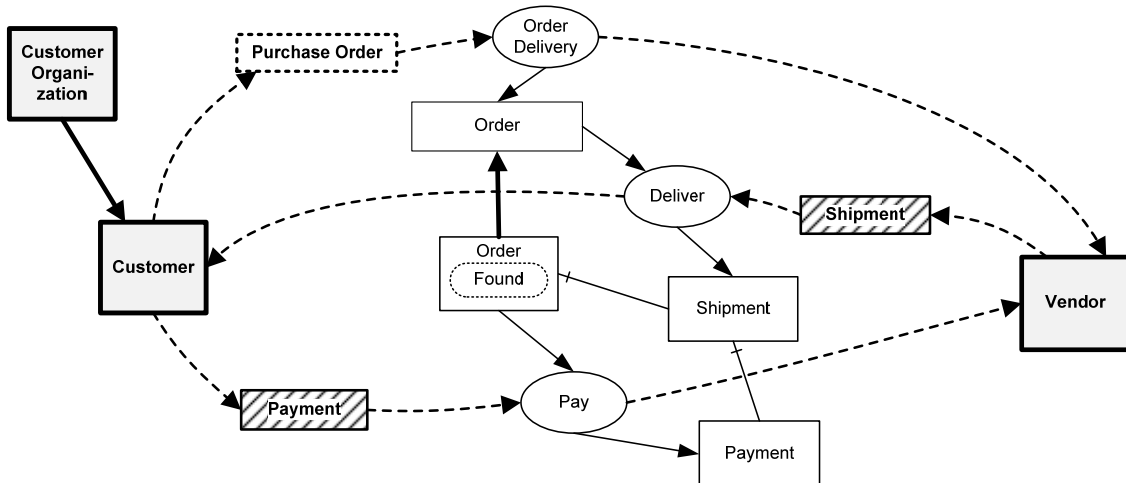


Figure 7: Derived and base interactions between Customer and Vendor.

is shown in the second part of the rule, represents derived interaction dependency. For instance, if $Order\ Delivery(Customer \dashrightarrow DB)$,

$Order\ Delivery(\perp) \rightarrow Order$, $(DB \rightarrow Vendor)$ and then $Order\ Delivery(Customer \dashrightarrow Vendor)$ where \perp represents an empty class. Two subsystems of Organization, DB (Database) and Vendor together with their interaction dependencies are represented in figure 6. This example is based on a well-known situation in Ford Motor Company after a radical change (Hammer, 2000). Ford Motor Company plays the role of an organization, which places a purchase order into a shared database (DB). The same service interaction loop, which was discussed previously, is represented in this diagram

as well. The interaction loop between Customer and Vendor represents an exchange of Shipment for Payment. Please note that the derived interactions cannot be in conflict with the specified dependencies in other diagrams. The interaction links, which are presented in figure 6, are consistent with the interaction dependencies of figure 7.

Static and dynamic similarities of active concepts can be shared by more specific concepts according to the following rule:

$$\text{if } X \Rightarrow Y \text{ and } Y \Rightarrow Z \text{ then } X \Rightarrow Z.$$

For instance, if a Company is a specialization of Customer Organization, and Customer Organization is an Organization, then for a Company can be

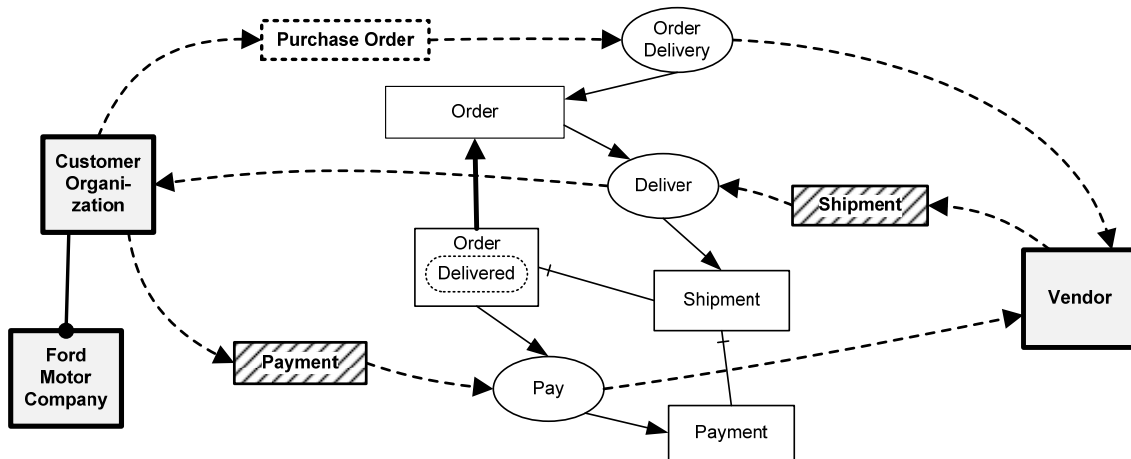


Figure 8: Derived interaction dependencies of Customer Organization.

applied static and dynamic similarities of an Organization.

More specific actors inherit interaction dependencies from the more generic actors. It should be noted that in the object-oriented approaches, inheritance link is defined just for attributes and operations. Inheritance dependency is convenient for sharing service interaction loops of more general actors. Interaction dependencies are inherited according to the following inference rules:

- 1) If $Action(Y \dashrightarrow Z)$, $Action(C1) \rightarrow C2$ and $X \blacktriangleright Y$ then $Action(X \dashrightarrow Z)$,
- 2) If $Action(Z \dashrightarrow Y)$, $Action(C2) \rightarrow C3$ and $X \blacktriangleright Y$ then $Action(Z \dashrightarrow X)$.

For example, if a Customer Organization is a Customer then Customer Organization inherits all service interaction links, which are represented for this more general concept. If $Order\ Delivery(Customer \dashrightarrow Vendor)$,

$Action(\perp) \rightarrow Order$ and $Customer\ Organization \blacktriangleright Customer$ then $Order\ Delivery(Customer\ Organization \dashrightarrow Vendor)$. Customer Organization has the opportunity to send a purchase order to a Vendor and Vendor is obliged to deliver Shipment to the Customer Organization. The derived interaction dependencies of Customer Organization are represented in figure 8.

Classification dependencies can be also used for reasoning about derived interaction dependencies between actors. Interaction dependencies are propagated according to classification dependency links. Interaction dependencies between actors are characterized by the following inference rules:

- 1) if $Action(Y \dashrightarrow Z)$, $Action(C1) \rightarrow C2$ and $X \bullet - Y$ then $Action(X \dashrightarrow Z)$,
- 2) if $Action(Z \dashrightarrow Y)$, $Action(C1) \rightarrow C2$ and X

$\bullet - Y$ then $Action(Z \dashrightarrow X)$.

For instance,

if $Order\ Delivery(Customer \dashrightarrow Vendor)$,
 $Order\ Delivery(\perp) \rightarrow Order$ and
 $Ford\ Motor\ Company \bullet - Customer$
 then $Order\ Delivery(Ford\ Motor\ Company \dashrightarrow Vendor)$.

This interaction loop can be replaced by simply switching from Customer to Ford Motor Company. It is represented in figure 9.

The responsibilities of different actors can be analysed using conceptual models of interactions. For instance, the Order Delivery action can be viewed as an opportunity to send a Purchase Order by Ford Motor Company to the Vendor. If Vendor accepts it, then he is responsible to Send Invoice to Ford Motor Company.

Please note that the opportunities, responsibilities, commitments and obligations of these two actors must be consistent with interaction dependencies. Inconsistency can be detected by naming the conflicts between actions or flows. More specific actors must be justified by their intrinsic communication actions, which are defined in terms of the complementary interaction dependencies of these actors. The presented inference rules are useful, but they are insufficient for reasoning about the consistency of interaction dependencies, which can be defined on different levels of specification. To understand the deep structure of service interactions, the behavioural and structural aspects of communication actions must be studied.

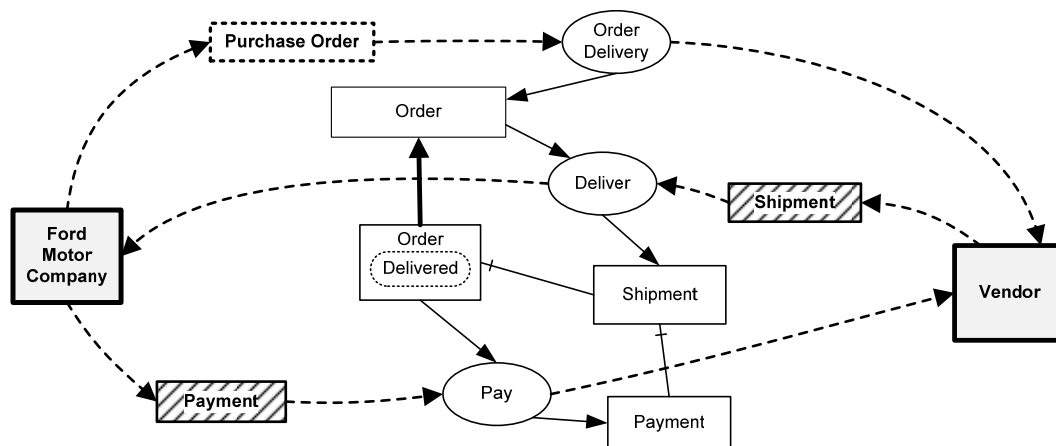


Figure 9: Derived interaction loop of Ford Motor Company.

5 CONCLUSIONS

The main contribution of this paper is presenting an integrated way of modelling. SICM provides us with a holistic method. One of the goals is to demonstrate how interactive, transitional and structural aspects of conceptual modelling can be integrated. Object-oriented modelling method projects static and dynamic aspects using different diagram types. In this case, to reach semantic integration of business processes and business data is very difficult. The semantic integration principles of different UML diagram types are not sufficiently clear. Since different modelling dimensions are highly intertwined, it is crucial to maintain integrity of various diagrams. We have demonstrated the interplay of three different aspects of conceptual models.

Interactions between actors are important to follow value exchange. Increment and decrement events represent economic resources exchanged in various business processes. Value exchanges are represented by two performing actions in opposite directions. The performing actions are triggered for the reason of coordinating actions. In this way, these actions are related to one value exchange. Every communication action is able to produce new facts that can be represented by various classes of objects. Value flow and service interactions provide the natural way of system decomposition. Value models help to clarify why enterprise actors want to exchange business objects with each other.

The ultimate goal of this paper is to overview deficiencies of conceptual modelling approaches and generic integration principles for development of holistic models of information systems. Bunge's

ontological principles of decomposition are lying in foregrounded in SICM method. Actors can be seen as organizational or technical system components. Organizational components are denoted by individuals, groups, or company divisions. Technical components can be seen as software or hardware system components. Decomposition of information system is based on semantic relations of classification, composition and inheritance. Similarities of these relations are explained in comparison with object-oriented approaches. Inference rules of the semantic relations are presented in this paper as well. The behavioural and structural dimensions of interactions were analysed in terms of creation, termination and reclassification action.

Conceptual modelling methods, which put emphasis on active concepts, typically focus on analysing interactivity between organizational and technical components. This tradition is quite successful for modelling of external behaviour of a system. In contrast, the object-oriented approach is based on modelling static and dynamic aspects of concepts, which can be represented by various classes of objects. The majority of textbooks in the area of systems analysis and design recommend concentrating first on domain modelling. Most conventional system analysis and design methods do not put into foreground modelling of active concepts. These methods project the structural, interactive and behavioural aspects into totally different types of diagrams that cause difficulties to integrate static and dynamic aspects of enterprise architecture dimensions. Very few emerging approaches to modelling make attempts to illustrate the deep interplay between active and passive

structures. We have illustrated with simple examples how to represent integration of various aspects of information systems.

REFERENCES

- Blaha, M., and Rumbaugh, J. (2005). Object-Oriented Modelling and Design with UML. London: Pearson.
- Bunge, M. A. (1979). *Treatise on Basic Philosophy, vol. 4, Ontology II: A World of Systems*. Dordrecht, Netherlands: Reidel Publishing Company
- De Marco, T. (1979). *Structured Analysis and System Specification*. New York: Prentice Hall.
- Dietz, J. L. G. (2001). DEMO: Towards a Discipline of Organisation Engineering, *European Journal of Operational Research (128)*, Elsevier Science, 351-363.
- Dietz, J. L. G. (2006). *Enterprise Ontology: Theory and Methodology*. Berlin: Springer.
- Dori, D. (2002). *Object-Process Methodology: A Holistic System Paradigm*. Berlin: Springer.
- Evermann, J. and Wand, Y. (2009). Ontology Based Object-Oriented Domain Modeling: Representing Behavior, *Journal of Database Management, Vol. 20, Issue No 1*, 48-77.
- Ferrario, R., and Guarino, N. (2008). Towards an Ontological Foundation for service Science, *First Future Internet Symposium, FIS 2008*, Vienna, Austria, 152-169. Berlin: Springer.
- Gane, C., and Sarson, T. (1979). *Structured System Analysis*. New York: Prentice Hall.
- Gemino, A. (1998). To be or maybe to be: An empirical comparison of mandatory and optional properties in conceptual modeling, *Proc. Ann. Conf. Admin. Sci. Assoc. of Canada*, Information Systems Division, Saskatoon, 33-44.
- Glinz, M. (2000). Problems and Deficiencies of UML as a Requirements Specification Language, *Proc. of the 10-th International Workshop on Software Specification and Design*, San Diego, 11-22.
- Guizzardi, G. (2007). Modal Aspects of Object Types and Part-Whole Relations and the *de re/de dicto* distinction, *19th International Conference on Advanced Information Systems Engineering*, Trondheim, Lecture Notes in Computer Science 4495. Springer.
- Gustas, R. (2010). A Look behind Conceptual Modeling Constructs in Information System Analysis and Design, *International Journal of Information System Modeling and Design*, Vol. 1, Issue No 1, 78-107.
- Gustas, R., and Gustiene, P. (2012). Conceptual Modeling Method for Separation of Concerns and Integration of Structure and Behavior, *International Journal of Information System Modeling and Design*, vol. 3 (1), 48-77.
- Gordijn, J., Akkermans, H., and van Vliet, H. (2000). Business Process Modelling is not Process Modelling, *Conceptual Modelling for E-Business and the Web*, LNCS 1921, 40-51. Berlin: Springer.
- Hammer, M. (1990). Reengineering work: Don't Automate, Obliterate, *Harvard Business review*, 104-112.
- Harel, D., Rumpe, B. (2004). Meaningful Modeling: What's the Semantics of 'Semantics'?, *IEEE Computer*, 64-72.
- Jacobson, I., and NG, P-W. (2005) *Aspect-Oriented Software Development with Use Cases*. Pennsylvania: Pearson Education.
- Lankhorst, M. (2005). *Enterprise Architecture at Work: Modelling, Communication, and Analysis*. Berlin: Springer.
- McCarthy, W.E. (1982). The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment. *The Accounting Review*, Vol. LVII, No. 3, 554-578.
- Nuseibeh, B., and Easterbrook, S. (2000). Requirements Engineering: A Roadmap. *In Proceedings of the Conference on the Future of Software Engineering: International Conference on Software Engineering* New York: ACM Press, 35-46.
- OMG. (2009) *Unified Modeling Language Superstructure, version 2.2*. Retrieved from www.omg.org/spec/UML/2.2
- Zachman, J. A. (1987) A Framework for Information System Architecture, *IBM Systems Journal*, Vol. 26, No 3.

Modelling Information Systems using NOMIS

An Overview of its Modelling Notation and Implementation

José Cordeiro

E.S.T. Setúbal, I.P.S, Campus do IPS, Setúbal, Portugal
jose.cordeiro@estsetubal.ips.pt

Keywords: Information Systems, Information Systems Modelling, Human-Centred Information Systems, Human Relativism, Organisational Semiotics, Theory of Organized Activity, Enterprise Ontology, NOMIS, NOMIS Vision, NOMIS Models, NOMIS Modelling Notation, NOMIS Metamodel.

Abstract: NOMIS is an innovative human centred information systems modelling approach that is based on human *observable* actions. Its goal is to achieve the desired objectivity and precision required to *engineer* information systems (IS). NOMIS proposes a vision into an IS from different views that are complimentary and comprehensive. Some of these views are adaptations and extensions of the theoretical IS insights provided by the theories of Organisational Semiotics, the Theory of Organised Activity and Enterprise Ontology. NOMIS also proposes a modelling graphical notation and a set of tables and diagrams to represent NOMIS vision and views. In this paper we provide a brief overview of NOMIS and its modelling aspects including NOMIS elements metamodel and NOMIS notation. A case study of a course system is used to show some practical examples of NOMIS notation application and also to deliver a real system according to NOMIS approach as one possible implementation.

1 INTRODUCTION

In spite many years of research and practice Information Systems (IS) are still developed with a loose understanding of information and how computers relate to human activities. From an information perspective, data as *information*, is stored in databases according to schemas developed without much business people intervention, or, otherwise, without objective guidelines for its conception. Many times these schemas are reproduced as user interface terms inadequately or not properly understood under the business context given the human nature of interpretation. Regarding human activities also often happens the supporting computer does not allow a human user for certain actions, or does not provide required information related to it.

NOMIS is an innovative human centred information systems modelling approach based on human *observable* actions that intends to improve modelling objectivity and precision. NOMIS proposes: (1) a vision composed by different views *inspired* by ideas from three known socio-technical approaches namely Organisational Semiotics (Liu, 2000), the Theory of Organized Activity (Holt, 1997) and Enterprise Ontology (Dietz, 2006a), and (2) its

visual representation composed by different models represented with a set of diagrams and tables. For this representation NOMIS provides its own modelling notation.

This paper extends and complements the work presented in Cordeiro, 2015. In this previous paper NOMIS vision was described and an empirical case study of a library system was used to highlight some modelling aspects and to show a practical application of NOMIS modelling approach. That paper shown several diagrams using UML profiles adapted from Cordeiro and Liu, 2007 and Cordeiro and Liu, 2008.

In this paper NOMIS elements metamodel and its modelling notation are presented together with a real case study of an e-learning system. This case study is modelled with some diagrams using NOMIS modelling notation. Furthermore, an implementation prototype putting into practice NOMIS Vision and ideas is given.

This paper is organized as follows: section 2 gives a brief overview of NOMIS vision together with NOMIS elements metamodel and modelling notation. Section 3 presents a case study and part of its modelling, section 4 introduces an e-learning platform and the e-learning prototype and, section 5 concludes and points some future research directions.

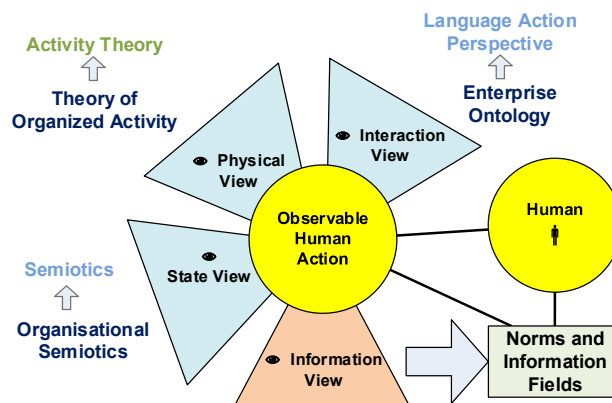


Figure 1: NOMIS Vision – its views and foundational theories.

2 NOMIS

2.1 Introduction

NOMIS – **N**ormative **M**odelling of **I**nformation **S**ystems is a human centred modelling approach to information systems development (ISD). NOMIS as a social-technical approach understands information systems “as human activity (social) systems which may or may not involve the use of computer systems” (Buckingham et al., 1987). Nevertheless its ultimate goal is the development of computerized systems suited for human use within organisations.

NOMIS kernel elements are *human actions* and *information*, both connected to the central human element, mandatory in any information system (IS). Human actions are present in many social approaches to ISD such as the Speech-act approach (see Hirschheim et al., 1997), based on language-acts as human actions, or the Activity Theory approach (Leont’ev, 1978) based on human collective actions, which are, both, in the roots of NOMIS.

Human actions used in NOMIS are human *observable* actions, those perceived by the human sensory system. This particular focus comes from its foundational philosophical stance – Human Relativism (Cordeiro et al., 2009) – that sees *observable reality* as “*more consensual, precise and, therefore more appropriate to be used by scientific methods*”.

The second kernel element in NOMIS is information, the basis of all information systems, still a misunderstood concept (see, for example, Falkenberg et al., 1996). NOMIS understands information as the result of an *interpretation* process coming after *perceiving* the *observed reality*. Following this idea, information is only available

from data after being interpreted by a human. There is no information without a human interpreter.

Information is the subject area of Semiotics which is the *study of signs* (see, for example, Chandler, 2002) where signs can be thought as information. In fact, Semiotics could be defined as the *study of meaning*: how meaning is created, represented, interpreted and communicated and meaning is all about information. Semiotics is also in the roots of NOMIS.

From a holistic view of human actions in general, NOMIS proposes a *vision* of information systems composed by a set of views addressing human interaction, action processes and context for actions inspired and based on, respectively, Enterprise Ontology (EO) (Dietz, 2006a), the Theory of Organized Activity (TOA) (Holt, 1997), and Organisational Semiotics (OS) (Liu, 2000). These views will be explained in the next section.

NOMIS views form a coherent and consistent vision of an IS from a human *observable* action perspective that is complemented with a fourth view related to *information* consumed, produced, stored and exchanged.

Considering the nature of human actions, NOMIS adds *Norms* as human behaviour regulators. *Norms* is a concept borrowed from OS (Stamper, 1996) that addresses and regulates sequences of human actions. Expected (human) behaviour is derived from *systems of norms* or *information fields* (IF) as they are called within OS (Stamper, 1996), where people tend to behave in a certain, expected and controlled way. Examples of IF are an organisation, a department, or even a family. IF and Norms are a *glue* connecting human actions and information.

NOMIS Vision is depicted in Figure 1.

Besides NOMIS Vision, NOMIS proposes a set of tables and diagrams and a modelling notation to

express and model IS that will be the basis for this work. A case study modelled and represented using NOMIS tables and diagrams can be found in (Cordeiro, 2015).

2.2 NOMIS Views and Vision

NOMIS Vision is the way NOMIS sees and understands IS. Its central element is the human and, in particular his/her *observable* actions. Focused in human actions it provides three different views or perspectives based on the theories of Enterprise Ontology, the Theory of Organized Activity and Organisational Semiotics as mentioned before. All of them were adapted and expanded in NOMIS. A fourth view addressing information is added. Each of these views will be briefly explained in the next subsections.

2.2.1 The Interaction View

The Interaction View covers the communicational dimension of human action. It is expected to model all forms of human communication and interaction within the IS. From these perspective besides looking at communication channels and interacting people also language-acts as seen in EO are modelled. Language-acts and any type of interaction acts may be represented by interactional patterns and reused in different contexts.

2.2.2 The State View

The state view uncovers and looks into environmental conditions or states that enable a human agent to act. It is concerned with context, state and state dependencies related to human actions. Its key element is the *environmental state* (ES) that is a composition of observable elements such as physical things (*bodies*) and/or information elements (an *information item* referred by its physical representation). The notion of ES is a NOMIS interpretation and adaptation of the *affordance* concept (Gibson, 1979 and, within OS, Stamper, 1996).

2.2.3 The Physical View

The physical view looks to material aspects related to human actions. A particular perspective addressed by this view is the representation of business processes showing (human) action sequences and activities. Also states and states transitions (driven by human actions) can be represented, which is a representation inherited from TOA.

The physical context is another aspect of the physical view that can be specified, for example, by locations (space and time).

2.2.4 The Information View

The Information view covers the information dimension of human action. Most of human actions depend or rely on information in different ways. There are some key assumptions NOMIS makes in this respect: (1) information does not exist without material support: a body or a human actor and, (2) information is created by humans or things (bodies) and consumed only by humans. From a human action perspective there is a focus on what information is required or consumed by the human performer, what information he/her has access and what information he/her produces. From a design perspective, it is useful to identify and model all information useful for a human action.

2.3 NOMIS Models

Models are used to represent simplified views of reality, capturing its essential elements according to a particular ontology. Models define a language and, as any language, affects the way world is perceived. NOMIS Models are just a way of representing NOMIS vision of IS reality. Following a Semiotic triadic sign model (Pierce, 1931-58) NOMIS vision is just a concept, a form of seeing an IS, and NOMIS Models one possible representation of NOMIS Vision as shown in Figure 2.

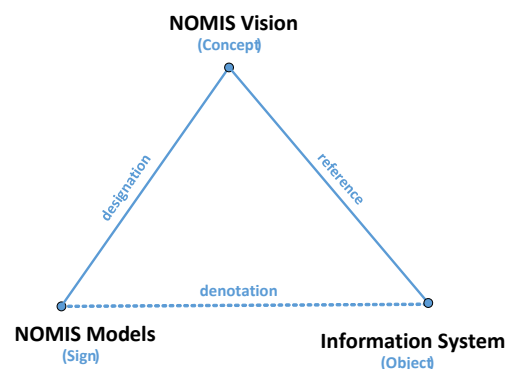


Figure 2: NOMIS Modelling Approach.

The *essential elements* represented in NOMIS Models correspond to the key concepts in NOMIS Vision and they are:

- Human Actions
- Actors – human performers
- Bodies – things

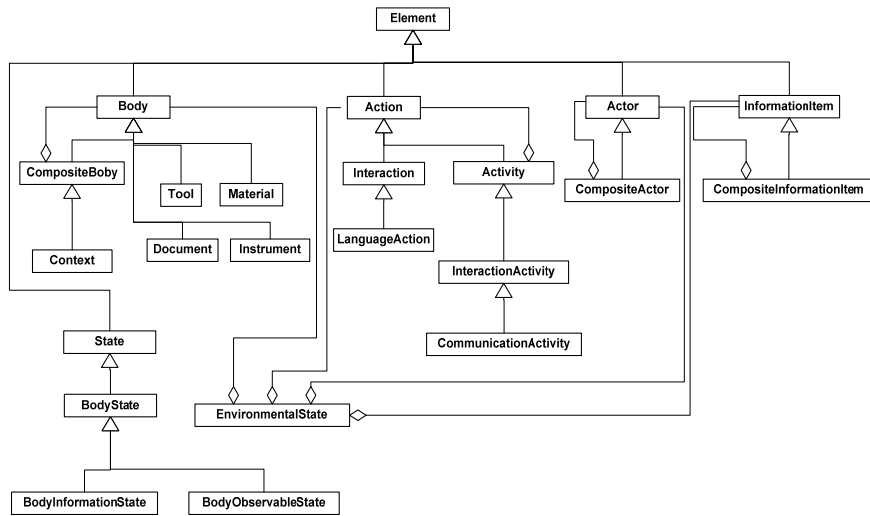


Figure 3: Metamodel of NOMIS Elements.

Table 1: NOMIS node elements notation.

Element	Symbolic annotation	Symbolic form	Element	Symbolic annotation	Symbolic form
Actor	A	○	Body	B	□
Action	AC	◯	Body State	BS	◻
Interaction	IA	↔	Composite element	C Initial letter	⊕
Language Action	LA	→	Environmental State	ES	◌
Information Item	I	▭			

- Information Items
- Language Actions (or Coordination-acts)
- Environmental States

It is also possible to have composite elements: a group of human actions as activities, a group of actors representing a team or organisation, or a composite of bodies or information items. A complete Metamodel of these *essential elements* is shown in Figure 3.

To represent NOMIS elements a modelling notation is provided in Table 1. Elements can be shown as rectangles with symbolic annotation, as symbolic forms or as appropriate icons.

3 MODELLING INFORMATION SYSTEMS WITH NOMIS: A COURSE SYSTEM CASE STUDY

This section will present a case study of a course IS

and will show a few models using the NOMIS modelling notation described in the previous section. There is no intent to provide the complete case study modelling but just a few aspects not covered in Cordeiro, 2015 and an illustration of some other aspects of the NOMIS modelling approach.

3.1 Case Study: An e-Learning System

This case study addresses a typical class course usually taught in licentiate degrees engineering courses at the School of Technology of the Polytechnic Institute of Setúbal. These courses run for a semester, having about 15 weeks of face to face teaching with one or more classes per week. Classes may be theoretical, practical, laboratories or theoretical-practical. Their duration varies between 1 hour and 2 hours. Apart from laboratories all classes take place in common class rooms equipped with a video-projector and a whiteboard. Before starting a course it is necessary to get information about enrolled students such as name, student number,

contact information, etc. and to prepare a few documents, namely a course form with information regarding course contents, evaluation method, bibliography and some other data. When a course is running, teaching is usually done face to face in a class room. It is common in these classes to give students some teaching materials such as presentation slides handouts, tutorials, articles, bibliography, exercises, etc. and some additional information such as deadline for exercises, evaluation schedules, extra class times, etc. Outside classes, teachers have attending hours for receiving students. Sometimes it is also necessary to contact students when a situation demands it: it may be something preventing a class to be taught or other out of the ordinary circumstances. In some courses, there is a project work involving 1 to 4 students and they must meet and collaborate outside classes. After classes are finished there are student evaluations, and resulting grades should be written in a form (obtained before) and delivered to the school secretariat. Also, it is necessary to fill a report per class type and per course concerning student attendance, grades, subjects effectively taught and other related information.

Computer support to teaching uses the school IS for official information such as course syllabus, student and teacher information, course schedules, etc. and Moodle for course related information such as course materials, communication between teachers and students, etc.

3.2 Applying NOMIS – First Steps

Although NOMIS Modelling approach doesn't propose a methodology a first step is to find its kernel elements: human actions and their performers. These elements will be collected using a model artefact named Human Action Table (HAT). HAT registers human actions, their intervening human actors and related elements such as bodies (things), *information items* and locations. In Table 2 there is a simple example of a course HAT. At this point it is not necessary to have a complete description, and missing elements may help to uncover important details. Some human actions identified correspond to general activities, e.g. "to teach", in this case action detail or atomicity will depend on model's purpose, such as to communicate, to design or to implement the system.

Table 2: Initial Human Action Table of a course.

	Human Actions	Initiator	Addressee	Bodies	Information Items	Local
1	To teach (face to face)	Teacher	Student	Slides, texts, pens		Class room
2	To attend class	Student				
3	To distribute document	Teacher	Student	Document		
4	To inform about something	Teacher	Student		Information	
5	To inform about something	Student	Teacher		Information	
6	To create exercise	Teacher		Exercise form	Exercise information	
7	To request exercise execution	Teacher	Student	Exercise form		
8	To do exercise	Student		Exercise form		
9	To submit exercise	Student	Teacher	Exercise form		
10	To evaluate exercise	Teacher		Exercise form		
11	To write course report	Teacher		Course form	Report information	
12	To produce course information	Teacher			Course Information	
13	To distribute course information	Teacher	Student		Course Information	
14	To attend students	Teacher	Student			

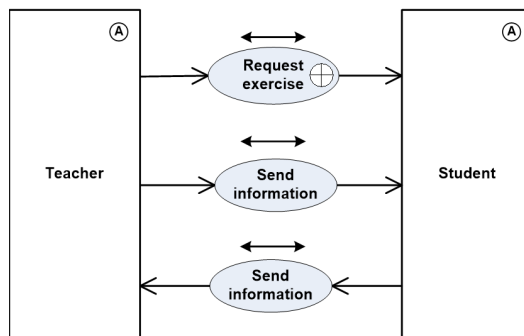


Figure 4: HID showing course interactions.

3.3 Interaction and Physical Views

As it is described this system doesn't have much elaborated action sequences, being composed mostly by interactions between teacher and students identified in the HAT by the initiator and addressee human elements. Two of these interactions are related to exercises requested by teachers and communication between students and teachers. EO models these kind of interaction using *business patterns* (Dietz, 2006b) understood as the fundamental building block for modelling any organisation at the ontological level. NOMIS is able to express these type of interaction patterns using Human Interaction Diagrams (HID) within the interaction view. In Figure 4 there is a HID showing both interactions. As in EO, in NOMIS the request exercise is depicted as a pattern – in this case a *composite interaction activity*. This means a composite action (expressed by the plus sign inside the activity symbol). This activity can be further decomposed using an action Sequence Diagram (ASD) from the physical view as an activity pattern. Usually, ASD diagrams are used to show typical business processes as a kind of UML Activity Diagrams but in this case it is used to show sequences of speech-acts as human actions (represented inside ovals). The emphasis here is the NOMIS ability to represent activity patterns as the one shown in Figure 5 where <work> may be replaced with different elements such as exercise, examination or even projects.

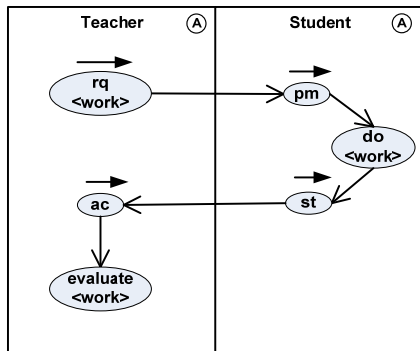


Figure 5: An ASD showing the request work pattern.

3.4 Information View

Although it is possible to show connections between human actions or *bodies* and information in specific NOMIS diagrams, e.g. (1) show all information required, auxiliary or produced by a human action or (2) show connections between human information

interpreters and respective body information carriers, here the emphasis is on how information is collected and represented. In this case, NOMIS uses an Information Items Table (IIT) where information items and its supporting bodies are described.

In Tabl some information items, required in the e-learning case study, are shown.

Table 3: Some course information items.

	Information Item	Information	
		Content	Related supporting bodies
1.	Enrolment Information	Student information	
	Description	<ul style="list-style-type: none"> • Student's name • Student number • Sex, date of birth • Contact 	
2.	Teacher Information	Teacher information	
	Description	<ul style="list-style-type: none"> • Teacher's name • Category • Title • Contact 	
3.	Course Information	Course information	
	Description	<ul style="list-style-type: none"> • Course name • Course • Semester 	

3.5 State View

NOMIS course state view is a key system view. It shows its fundamental environmental states (ES) providing an overall system perspective. Business processes are just *paths* between ES or *paths* to achieve an ES. Some ES may be understood as *goals* as they correspond to desired states such as "to complete a course". ES are shown in NOMIS using Environmental State Diagrams (ESD) where links between the different action states represent existential dependencies. In Figure 6 it is depicted a course system EDD where a running course (an ES) is dependent on teacher and student ES. These ES on the other hand are dependent, respectively, on teacher information (an information item) and person (a person *body* as teacher) and enrolment information and person (as student).

3.6 Norms and Information Fields

A last and important element to be modelled according to NOMIS is *norms*. Many of these norms can be extracted from NOMIS diagrams such as some norms regarding action sequences, existential dependencies, and information required or auxiliary

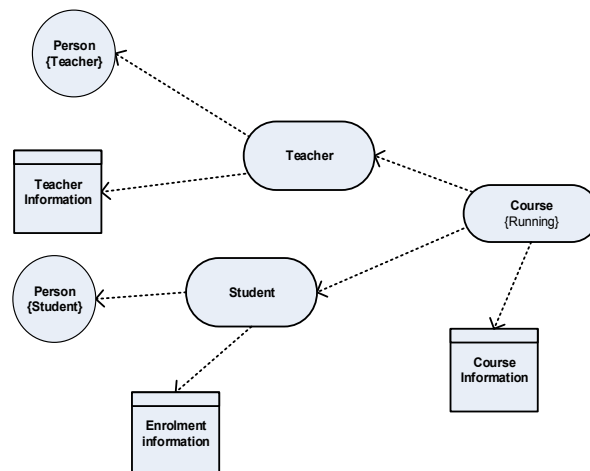


Figure 6: Course system EDD.

To actions among others. A small list of course system norms for illustrative purposes are:

- Classes Scheduling;
- Semester Calendar;
- Teacher attending hours;
- Teachers responsibility to create class summary, course evaluation rules, bibliography and theoretical contents;
- Teachers responsibility to produce course reports, and to send evaluation information to the school;
- General school evaluation rules

Some of these norms may just be used as information in the context of particular human actions or, for example, be incorporated in the model.

In the course system context there are different norm systems or information fields (IF). These are the school, the department and the course. Each of them may use its proper terms and have some particular rules. For example a student may not attend a course unless it is a registered student according to the school norms but, sometimes due to a possible delayed registration process it is authorized within the course scope (IF) to attend it without being legally registered.

4 IMPLEMENTING AN E-LEARNING SYSTEM ACCORDING TO NOMIS

The previous section provided a short and simple analysis and modelling of a course system. In this section, our goal is to present a simple implementation of an e-learning system to support

that (*human*) course system. This application should be seen as one practical application of NOMIS, still many other forms of using NOMIS in practice are possible.

The implemented e-learning system is a computer system that will be used to support some of the course human actions modelled using NOMIS. Its simple use will be as a repository for class materials such as texts, documents, presentation slides, etc. and also as a communication tool that will enable information exchange among participants. A consequence of using this supporting system is that human actions will change, for example “to request an exercise”, a physical action of giving a piece of paper to a student will be replaced by a menu entrance that supports this action by sending an electronic document to that student. This change is often neglected when implementing business processes for example. Therefore, there will be new human actions for the e-learning system such as to store a document, to send a document, to retrieve a document, to send a message, to retrieve a message, to send and store a document, to view a document, etc. An advantage of designing a system from this perspective is that each relevant human action can be individually analysed from a business oriented view and its computer support can be furnished appropriately. Also, effective user needs can be fulfilled accordingly. Besides furnishing support for specific human actions the system can also help by giving useful information related to those actions such as how to execute them, norms affecting those actions, or available tools. A separate awareness system may be designed and implemented with this purpose.

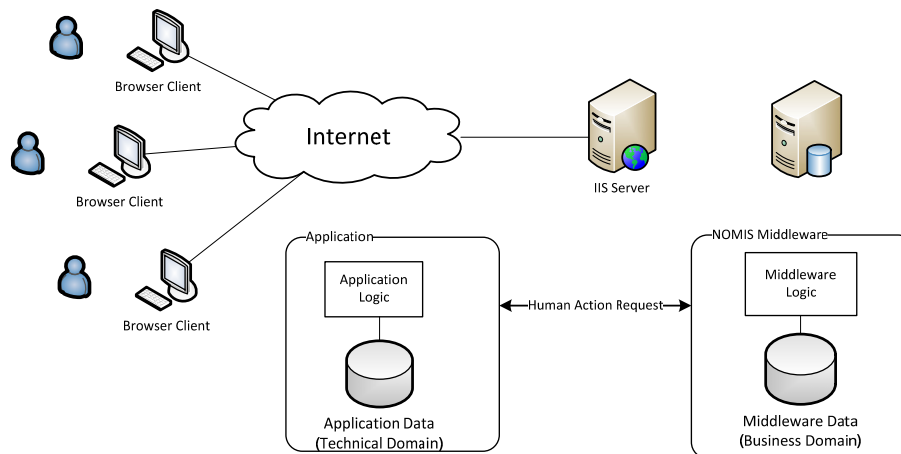


Figure 7: The e-learning platform architecture.

4.1 NOMIS Platform Architecture

For implementing the e-learning system a basic Client-Server based architecture using the Internet was chosen (see Figure 7). However, this architecture was further adapted to be aligned with NOMIS concepts. Accordingly, there are two separate modules: (1) to handle application specific aspects and (2) to handle NOMIS related features. The last one is a kernel middleware – NOMIS middleware – used to provide support for user actions, user information and user communication functionalities. NOMIS middleware includes a relational database to support business data and a logic layer responsible to handle requests from applications. The application module, on the other hand, handles most technical aspects, including presentation logic, interaction with NOMIS middleware, technical parts of application logic and may include its own database of technical data. Separation of technical and business aspects is an essential characteristic of this platform and its based applications. This separation is accomplished by assigning to NOMIS middleware the management of any element seen as part of the business domain.

This architecture is not specific of the e-learning application and can be used by any other NOMIS application.

4.2 NOMIS Middleware

NOMIS middleware (NOMIS MW) is an independent layer responsible for connecting an application to general business information and human action supporting features. Basically it consists of a database composed by a group of tables that store all important business related information and a logic layer responsible for managing access to

it. NOMIS MW tables store NOMIS elements and relationships between them. There is a table per each element: Person, Action, Body, Activity and Role. Role in this case represents a “person state” and for each valid relationship between elements: Action-Body, Person-Body, Body-Body, Role-Action, Role-Person, Activity-Role and Activity-Activity.

A distinctive feature of NOMIS MW tables is that all tables, except for relationship tables, contain a group of similar columns having the following structure:

ID	GUID	Name	Description	StartTime	FinishTime
----	------	------	-------------	-----------	------------	-------

Bold column fields represent required data and cannot be empty.

This table structure is partially inspired by the Semantic Temporal Database (STDB) proposed by OS (Liu, 2000). In fact, it keeps its temporal dimension by having a start and finish time for each element, allowing it to change.

The GUID field stores a global unique identifier that is attached to each record and table. It is possible using this strategy to change completely the information about a person, for instance using a different information table, as the GUID will stay the same the person information will be relative to its existing time period.

4.3 The e-Learning Prototype

A prototype using NOMIS Platform architecture was developed using Microsoft .NET platform, a screen shot is shown in Figure 8. Some notes regarding this implementation:

1. The school, each course and each class is defined as an activity giving a context for actions.

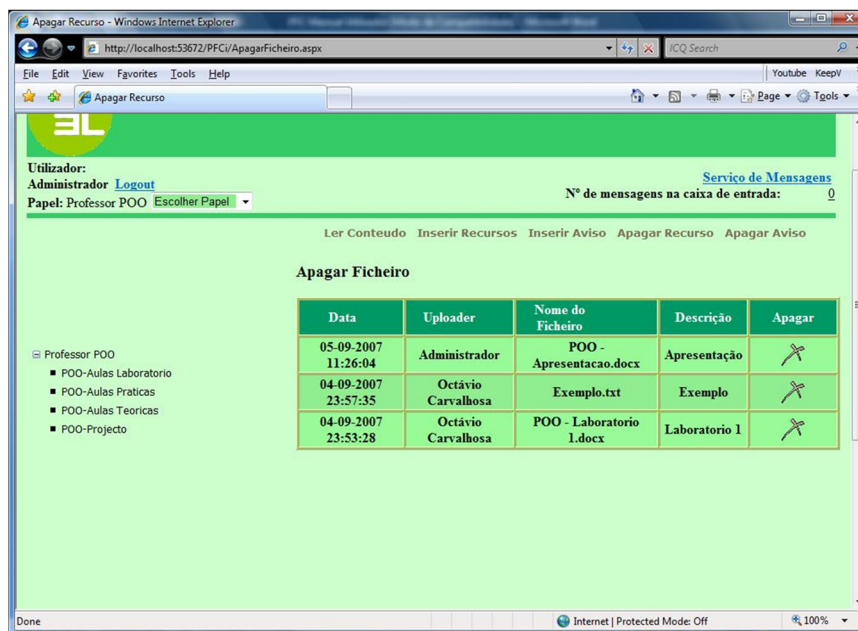


Figure 8: Screen shot of the e-learning application prototype.

- Information Items are supported and created as tables and records in the database, or otherwise as documents referred also by records in the database.
- Three roles were created, namely administrator, professor and student.
- Actions correspond to ASP.NET pages triggered by menu selections, buttons or hyperlinks.

NOMIS MW does not deal with visualisation and interaction aspects; those can be designed by a designer and implemented using application specific visualisation controls. From an application point of view, NOMIS MW can furnish all actions available for a certain activity, taking into account its user, associated role, any applicable norm and all contextual information stored on bodies. This separation helps keeping the design independent of many application technical details.

All the necessary information about available activities, action, and bodies is provided through NOMIS MW.

5 CONCLUSION AND FUTURE WORK

This paper presented a brief overview of NOMIS and its modelling aspects including NOMIS elements metamodel and part of NOMIS notation. A case study of a course system was used to show some practical

examples of NOMIS notation application and also to deliver a real system according to NOMIS approach as one possible implementation. The e-learning prototype implemented took advantage of a NOMIS specific infrastructure that could be used for other NOMIS applications.

NOMIS modelling approach is fully described in (Cordeiro, 2011).

Regarding future work there is much to do to validate and test NOMIS modelling approach. A new prototype is necessary to uncover additional modelling and practical aspects. One possibility is the use of Model-Driven Engineering to produce different applications using some ideas from the created prototype infrastructure. In this respect, NOMIS metamodel and the modelling notation can be used to create, respectively, the abstract and the concrete syntax of a Model-Driven approach.

REFERENCES

- Buckingham, R. A., Hirschheim, R. A., Land, F. F., and Tully, C. J. (eds) (1987). Information systems curriculum: A basis for course design, *Information Systems Education: Recommendations and Implementation*. Cambridge University Press, Cambridge.
- Chandler, D. (2002). *Semiotics: The Basics*, Routledge, London, UK.
- Cordeiro, José A. M. 2015. A New Way of Modelling Information Systems and Business Processes - the

- NOMIS Approach. In *Business Modeling and Software Design*, ed. Shishkov, Boris, 102 - 118. ISBN: 978-3-319-20051-4. Berlin, Heidelberg: Springer International Publishing.
- Cordeiro, J. (2011). *Normative Approach to Information systems Modelling*. PhD Thesis. The University of Reading, UK.
- Cordeiro, J. and Liu, K., (2007). UML 2 Profiles for Ontology Charts and Diplans - Issues on Meta-modelling. In *Proceedings of the 2nd International Workshop on Enterprise Modelling and Information Systems Architectures*. St. Goar, Germany.
- Cordeiro, J. and Liu, K., (2008). A UML Profile for Enterprise Ontology. In *Proceedings of the 2nd International Workshop on Enterprise Systems and Technology*. Enschede, the Netherlands.
- Cordeiro, J., Filipe, J. and Liu, K., (2009). Towards a Human Oriented Approach to Information Systems Development. In *Proceedings of the 3rd International Workshop on Enterprise Systems and Technology*. Sofia, Bulgaria.
- Dietz, J., (2006a). *Enterprise Ontology, Theory and Methodology*. Springer-Verlag, Berlin Heidelberg, Germany.
- Dietz, J. (2006b). The Deep Structure of Business Processes. In *Communications of the ACM*, 49, 5, pages 59 -64
- Falkenberg, E., Hesse, W., Lindgreen, P., Nilsson, B., Oei, J., Rolland, C., Stamper, R., Van Assche, F. Verrijn-Stuart, A. and Voss, K. (1996) *FRISCO : A Framework of Information System Concepts*, The IFIP WG 8.1 Task Group FRISCO, December 1996.
- Gibson, J. (1979). *The Ecological Approach to Visual Perception*. Houghton Mifflin, Boston.
- Hirschheim, R., livari, J., and Klein, H. (1997). A Comparison of Five Alternative Approaches to Information Systems Development. In *Australasian Journal Of Information Systems, Vol 5, No. 1*.
- Holt, A., (1997). *Organized Activity and Its Support by Computer*. Kluwer Academic Publishers, Dordrecht, The Netherlands.
- Leont'ev, A. (1978). *Activity, Consciousness, and Personality*. Prentice-Hall, Englewood Cliffs.
- Liu, K., (2000). *Semiotics in Information Systems Engineering*. Cambridge University Press, Cambridge, UK.
- Peirce, C. S. (1931-58). *Collected Writings* (8 Vols.). In Hartshorne, C., Weiss, P., Burks, A. (eds). Harvard University Press, Cambridge, MA.
- Stamper, R. (1996). Signs, Norms, and Information Systems. In *Signs of Work*, Holmqvist B. et al (eds). Walter de Gruyter, Berlin.

Hard Problems and Soft Information

Coen Suurmond

*RBK Group, Keulenstraat 18, 7418 ET Deventer, The Netherlands
csuurmond@rbk.nl*

Keywords: Business Processes, Process Logic, Organisational Semiotics, Sales & Operations Planning.

Abstract: Finished product planning is a key business process for companies. It is about finding the balance between service levels and cost, and is therefore critical for the success of the company. In this paper the structure of the problem will be analysed and compared with literature about sales & operations planning as well as ERP solutions. In the analysis general process logic will be contrasted with idiosyncratic characteristics of the individual company. The use of different kinds of information will be discussed, in combination with the formal sign system of the computer and the social sign system of human communication.

1 INTRODUCTION

Finished product planning is a key business process for companies. This planning decouples demand from production and it is a highly determining factor both for market volume and for profit margins. On the market side customer service level and lead time are key for market share while on the production side it is crucial to keep down variable costs. In this paper a specific product group (fresh meat) for a specific market (multi-store retail chains) will be used to analyse the business processes and information flows that are in play, and it will be examined how this problem is covered in literature and by ERP software.

In this paper the paired concepts of process logic / idiosyncrasy will play an important part. Process logic will mark the necessary general structure underlying the actual business processes of the individual company. This structure will always be present in every business in a particular market, because certain structures are inherent to and inevitable in operating with these products on those markets. Alongside this there are the idiosyncratic characteristics of the individual company. Companies after all do differ, even though they operate with the same products in the same market segment. The individuality of the company is the foundation of the existence of any company on the market. As John Kay pointed out, the distinctive capabilities of the company are what distinguishes the company from its competitors and that form the foundation for the success of the company. And, again as pointed out by John Kay: "A firm can achieve added value only on

the basis of some distinctive capability – some feature of its relationships which other firms lack, and cannot readily reproduce" (Kay, 1993, p. 64).

The analysis of business processes in terms of process logic and idiosyncratic characteristics has two aims. Firstly, it is a method to map business processes in a way that helps external consultants (specialists in general patterns) and internal employees (specialists in specific details) to come to a mutual understanding and a common basis. Secondly, it is about the awareness of the intangible, the understanding that not everything can be reduced to schemas and fixed rules. Information systems should not be a goal onto themselves but rather serve to improve the market position of the company, which they must do by adequately supporting the business processes. The process logic provides insight into the general structures, while the idiosyncratic characteristics provide insight into the way in which the processes actually happen within the company (including the often blurred boundaries between processes). This approach should also help in getting a feel for the distinctive capabilities of the company, competitive strengths that must be preserved and possibly enhanced in developing a new business information system.

Besides the paired concepts of process logic / idiosyncratic characteristics the analysis of the nature of the information in business processes with relation to formal and social sign systems used forms a second pillar of this analysis. Computer systems are formal sign systems, highly capable in processing declarative information, but they have trouble with vague

boundaries, “unclean” categorisation and weighing heterogeneous norms that cannot be fulfilled simultaneously against each other (delivery in full but slightly delayed or delivery on time but slightly less than ordered? Delivery should be at the right time in the right quantity with the right quality, which ‘right’ might be relaxed in which context?). Social sign systems are much better in dealing with meaning in context, modalities, and intentions (discussed in Suurmond, 2015). In the creation of a business information system awareness of the nature of the information and the conscious choice for the right sign systems is critically important for the effectiveness and efficiency of the business processes. Of the coordination mechanisms identified by Mintzberg direct supervision and mutual coordination are much more based on informal sign systems, while the application of formal sign systems presupposes standardisation (Mintzberg, 1979). The effectiveness and efficiency of business processes is of course dependent on correct, timely and complete information (Starreveld, 1963), but also dependent on the degree to which the available information is relevant and accessible (Grice, 1989).

The paper consists of six sections. After a short introduction to the problem area, the business processes involved are analysed in terms of process logic and idiosyncrasy. This is followed by an analysis of the nature of the information used in the business processes, and the role of formal and social sign systems in conveying the information. Literature about sales and operations planning is discussed in section five, and the paper ends with a recapitulation and conclusions.

2 PROBLEM AREA

This paper is concerned with the finished product planning for the production unit for pre-packaged meat in retail chains. A typical production unit produces some 100 to 200 different fresh pre-packaged meat products on a daily basis. Incoming shop orders need to be made available ready-to-ship on the loading dock for transport, delivery reliability must be above 99.7%. The products have an internal shelf life of at most two days. Everything is produced from fresh ingredients each day. And, of course, waste and production costs must be kept to a minimum.

Many products have a fairly stable demand pattern. Demand is however highly irregular in case of promotions, product introductions (which do not have a demand history) and a number of articles that

are weather-dependent. Products which have been part of a promotion in the past weeks and seasonal products also have demand irregularities.

Planning is generally done in a large Excel in which the order history over the past weeks for each distribution timeslot is recorded along with the demand prognosis for promotions and in which the planner records the amounts per product per day that need to be produced. This last list is processed further within the production planning to create production orders as well as lists for the resource, man-hour and line-hour needs. Each business within this sector has developed its own particular solutions over time, the common characteristic is the use of Excel with order history as input and production lists as output. The expected demand for promotions is mostly determined in a separate process and then made available to the planner.

3 PROCESS LOGIC & IDIOSYNCRASY

3.1 Process Logic

Every company that produces for a market in which the lead time between order time and delivery time is less than the time necessary to produce the goods will work with a (semi) finished product stock. This stock must be sufficient to fulfil the orders within the delivery reliability requirements in this market. Further, the stock has to be as low as possible because of warehouse costs and to minimise waste. Finally, production will have requirements regarding the frequency and size of the production batches. All of this leads to the following three processes that will always be found (but, more often than not, implicitly rather than explicitly):

1. Determining expected demand
2. Determining the target stock level
3. Determining the production output

The distribution pattern defines loading times for groups of orders, before the scheduled loading time the orders must be picked and made available at the loading docks. Therefore, distribution and production is organised in timeslots, in each distribution timeslot a set of orders is picked, and in each production timeslot a set of products is produced. Shops have a timeslot for ordering. Expected demand is specified per distribution timeslot, which would correspond with one or more ordering timeslots. The end times of ordering timeslots and distribution timeslots are

fixed, the end time of production timeslots is more flexible.

Given the daily delivery to the shops in combination with the internal shelf life of two days at most, the full product range will be produced and distributed in a 24 hour cycle. For each day and each timeslot expected demand will be estimated, target stock levels will be set and production output levels will be set. The planning moments are:

1. After each ordering timeslot
2. Just before the production day
3. Just before the production week
4. A few weeks before the production week
5. About four to six weeks before the production week

After each ordering timeslot the planning is checked and adjusted where necessary (and possible), one day in advance production orders are generated and fresh raw materials are ordered, the week before the production week production schedules are set and suppliers will be informed about expected demand, and the same applies for the planning moment a few weeks before the production week (same information, less certainty. In the first planning moment about four to six weeks in advance of production promotions are planned (because of the increased quantities it is important to have agreements in place with suppliers regarding price and expected demand of raw materials).

3.2 Idiosyncrasies in Individual Companies

Although the processes described above in the process logic can be found in each company in this market, in organisation and actual execution a great variety exists. Often, boundaries between the processes are blurred, both organisationally and in the Excls used for planning. Much of the knowledge and information is personal and non-coded (Boisot, 1998). Many problems are either 'spirited away' by some creative and experienced old hands, or solved in informal communication. Sometimes, this is a good thing, because the problem solving capabilities of the company are much greater than one would expect from studying organisation charts and documents about process flows. Sometimes, it is good but too vulnerable because of the dependencies on one or two key figures in the organisation. Sometimes, it is bad because problems are not really solved, only moved out of perception.

4 NATURE OF INFORMATION & SIGN SYSTEMS

4.1 Information for Determining Expected Demand

In the process 'determining expected demand' (forecasting) the single information product appears to be a list with the expected shop orders grouped by timeslot and by saleable item, and the primary source of information is demand history. However, although extrapolation from history to expected demand may lead to a convenient list of hard data, it does not tell the whole story. Firstly, demand history is never the only source of information. In case of promotions and product introductions no relevant and reliable demand history is available (although, the history of promotions and product introduction might offer useful indications), and demand for some products might be dependent on future conditions (weather) or situations (events, bank holidays). Other information sources are needed, and the information from these sources must be interpreted in context. The interpretation of the weather forecast, especially in case of a possible sudden change of the weather, is an example. The assessment of the impact of publicity (kind and scale) in case of promotions and product introductions is another example. Secondly, expected future demand is never a single value, but rather a spread. If you were to discuss expected demand for a certain product with a group of experts, they would say that demand is expected somewhere in the range between X and Y.

4.2 Information for Setting Stock Levels

Given the outcome of the process 'determining expected demand', the job of the stock planner is to set the target levels such that stock will be sufficient when demand is at the maximum level of the range, and stock will not be wasted when demand is at the minimum level. Depending on the volatility of demand and on the allowed storage life of products in stock, the planner can have an easy job (steady demand, longer internal storage life) or an impossible job (highly volatile demand, very short storage life). Impossible situations are not primarily the problem of the planner however, especially if this is a recurring issue. It is a problem of setting realistic norms for planning, probably for someone higher up in the organisational hierarchy.

It gets interesting in the border area when the

planner has a difficult but doable job. In this area the planner must be creative and use all his available knowledge and information sources. The planner might collect further information from experienced demand experts in order to reduce the expectation spread, or organise extra production capacity in order to react quickly (quicker than normal production schedules would allow) in case of impending stock shortages. The planner will juggle with delivery risks and production reaction times in order to find his solution. Background information about what was possible or impossible in comparable situations in the past and informal discussions play a major role in this kind of decisions.

4.3 Information for Setting Production Levels

The third process must set production levels such that production efficiency is optimised. Lot size and production capacity are the primary determining factors. Both factors represent discontinuity. Often, products (or semi-finished products) are optimally produced in fixed amounts, due to the capacity of machinery. Optimal operation times for production lines vary in units of x hours (8 hours is a typical value), due to work schedules (people work in shifts). The combination of lot size and optimal operation times will lead to a production mix that satisfies the minimum and maximum stock levels set in the preceding process. In standard situations this can all be calculated according to fixed patterns and decision rules. In some situations not all requirements can be met at the same time. In these cases additional information is needed, either in finding ways to squeeze a little more out of production, or in making sure that some products will be produced in the required quantity, or in assessing the weight of the different norms in the given situation and accepting the additional risk or additional cost.

4.4 Sign Systems

Considered from the viewpoint of the process logic as analysed above, the information about expected demand, stock level planning and production-level planning seems pretty straightforward and a perfect fit for the domain of formal sign systems. Essentially it is about three consecutive datasets with the same structure: date, timeslot, item code, quantity. In demand this dataset represents an expectation, in stock-level planning and in production-level planning the dataset represents a target.

Considered from the more detailed analysis of the

information actually used in real planning situations, it will be clear that other kinds of information and other kinds of sign systems are involved. The inaccuracies of demand expectation, actual stocks and actual production output must be dealt with; the planner works with patterns based on experience and history; risks and possibilities are discussed between planning, sales and production; and accompanying instructions are given to the operators in the shop-floor processes. Information takes the form of background knowledge, consulting colleagues, oral communication, and written notes.

4.5 Idiosyncrasies Revisited

Companies differ from each other in the way they execute their key processes, and for the production of fresh food for retailers the finished product planning is such a key process. In this planning process success on the market (delivery reliability, lead times) and internal success (minimising production costs) come together. The way formal sign systems and informal sign systems are used to support this difficult and critical process is highly characteristic for each individual company. Depending on the distribution of knowledge and experience in one company the 'real' decisions and adjustments might be made by production management (using demand and stock information), and the planner is no more than a rather passive Excel-driver and data cruncher. In another company, the planner plans and the shop floor executes. In a third company, production and management meet each day in order to prevent upcoming problems and smooth out existing problems. Each company can either flourish or be ailing. It all depends on the quality and the fit of the information to and from the planning process and on the quality of the persons who make the decisions.

5 SALES & OPERATIONS PLANNING

The subject of the case is the coordination between production and expected demand using finished product planning, an area termed Sales & Operations Planning in the literature and for which ERP systems provide support. It is then useful to look at what the literature says about this and to what extent literature can support the analysis of the case. First a definition from internet: "Sales and Operations Planning (S&OP) is an iterative business management process that determines the optimum level of manufacturing

output”. This definition fits the theme of this paper, although the term ‘optimum’ in the definition is an empty shell without criteria and it is a planning process rather than a management process. The definition proceeds to state that “The process is built upon stakeholder agreement and an approved consensus plan. To help stakeholders agree on a plan of action based on real-time data, S&OP software products include dashboards that display data related to equipment, labour, facilities, material and finance. The purpose of the dashboards is to provide the stakeholders with a single, shared view of the data”. This is not true in the situation considered here. The planner has a delegated responsibility to solve the planning problem within the set norms and to signal when he is structurally unable to meet the norms. Occasional deviations are permitted (and delivery reliability prevails over costs), structurally both norms need to be met. Determining tight but achievable norms is a mutual undertaking in which all stakeholders are involved and in which at least commitment, if not consensus, needs to be achieved. Operational planning is very different in nature. This holds even more strongly if Sales & Operations Planning is not done on a monthly basis, but, as in the case, must be done on a weekly and daily basis.

The definition in the APICS dictionary, which should be an authoritative source given the status of APICS as an organisation (“the premier professional organisational for supply chain management”, according to its website, with over 43000 members and more than 300 international partners), provides even less of a guide. Because of the language used I will cite the very long lemma in full:

(APICS Dictionary, 2008, p.121f)

“Sales and operations planning – a process to develop tactical plans that provide management the ability to strategically direct its business to achieve competitive advantage on a continuous basis by integrating customer-focussed marketing plans for new and existing products with the management of the supply chain. The plan brings together all the plans for the business (sales, marketing, development, manufacturing, sourcing, and financial) into one integrated set of plans. It is performed at least once a month and is reviewed by management at an aggregate (product family) level. The process must reconcile all supply, demand, and new-product plans at both the detail and aggregate levels and tie to the business plan. It is the definitive statement of the company’s plans for the near to intermediate term, covering a horizon sufficient to plan the resources and to support the annual business planning

process. Executed properly, the sales and operations planning process links the strategic plans for the business with its execution and reviews performance measurements for continuous improvement. See: aggregate planning, production plan, production planning, sales plan, tactical planning.”

This is a definition (or description) of everything and therefore of nothing. Why should Sales & Operations Planning not be about the common daily operational practice of coordinating demand and availability and about no more than that? What does something like “a process to develop tactical plans that provide management the ability to strategically direct the business ...” add to our understanding of the problem?

Donald Sheldon writes in his *World Class Sales and Operations Planning* (co-published with APICS) “The S&OP process can have a major impact on the management of inventory” (Sheldon, 2006, p. 29). He then devotes chapters to “Creating the Demand Plan” and to “Operations Planning for the S&OP Process”. For Sheldon the S&OP process is the coordination between the various subplans (“Stated in its simplest terms, the S&OP process is a monthly planning cycle where plans for both customer expectations and internal operations are reviewed for accuracy, process accountability, lessons learned, and future risk management”, Sheldon, 2006, p. 2), where it should be essentially about the planning process itself. Of course there is an important role for higher level long term planning in companies to coordinate market developments, production capacities and resource needs. In this kind of higher level coordination operational norms must also be determined and adjusted, and possible measures should be agreed upon to ‘land’ changed norms with the relevant internal and external stakeholders. Donald Sheldon recognises the subordinate role of software: “All that is needed is a spreadsheet and good problem-solving tools and skills” (Sheldon, 2006, p. 15). The question remains, however, where the information for this problem solving will come from, and how to organise the different kinds of information flows (both formal via systems and informal via humans).

Robert Davis analyses what he calls the push-pull hybrid for supply chain management. “This hybrid model is based on the premise that you push produce and pull distribute” (Davis, 2016, p39). This analysis matches what was described above as the structure of the problem. His further analysis concentrates on what is happening in the supply chain as a whole. The chapter about inventory optimisation discusses the development of inventory policies that can be

translated into algorithms and executed automatically. This approach does address the problem of how to develop ways of coping with the problem of inventory levels, but it does not address the problem of the individual planner who uses information and who makes decisions.

Shaun Snapp (S&OP in Software) describes the standard S&OP process as follows: (1) review and sign off the demand plan; (2) review and sign off the supply plan, and (3) review and sign off the financial plan. And he gives the time features as a planning horizon between 1 and 5 years, a monthly planning frequency, and a monthly planning bucket. This is not quite the horizon the companies discussed here are working with. Planning of inventory levels is not in his list of plans to sign off, Snapp discusses dynamic safety stock in the chapter entitled ‘How Misunderstanding Service Level Undermines Effective S&OP’. He writes: “Safety stock is often set in companies by simply allowing individuals to guesstimate what the safety stock value should be and then provides them with the rights to make the safety stock adjustments” (Snapp, 2016, p. 105. This is followed by the remark that in his experience he never saw this working well, and “Stock levels should not be controlled by manually adjusting the safety stock. Instead, safety stock should be dynamically calculated and automated, and only changed as a result of changes in the variability of supply or demand” (Snapp, 2016, p. 107). The last part of course is true (from ‘only changed as...’), but the first part presupposes that variability of demand is represented perfectly in the computer system, with all relevant information taken into account. This clearly cannot always be true. And, if manual adjustment of demand forecast is allowed, planners very quickly learn the trick how to adjust demand in order to get the safety stock level they want.

The ERP systems of course offer solutions for S&OP. Hamilton in his book about MS Dynamics AX, paragraph 10.1: “You can automatically calculate the safety stock requirement based on variations in historical usage and the desired customer service level” (Hamilton, 2016, p. 236). A bit further in the same chapter, in paragraph 10.7: “When using the min-max coverage code, you specify the item’s minimum quantity and maximum quantity for each relevant site/warehouse. The minimum quantity represents the average daily usage multiplied by the item’s lead time” (Hamilton, 2016, p. 253). So, you either have safety stock, taking variation of demand into account; or you have a min-max policy where average demand represents the minimum stock needed? Dickersbach gives in his book “Supply

Chain Management with SAP APO™” the following structure of the demand planning process (somewhat shortened in my representation): (1) Forecast; (2) Check on plausibility of the forecast; (3) Production planning. Inventory planning is not mentioned at all. (Dickersbach, 2009)

A picture of confusing, vague and contradicting terminology in combination with conceptual weaknesses arises from the works mentioned above. Neither in the literature nor in the software clear structures of the problem area are defined. A concrete example of the translation of this messy approach into actual customer requirements is the following set of questions for candidate software suppliers by a company in the food industry:

- “How will stock adjustments automatically influence production schedule?”
- “Sequence of production is determined by the scheduling process. Disruptions in other processes (up and down) lead to automatic rescheduling of production capacity; manual adjusting to schedule needs to be validated and recorded as an exception”
- “How will the production schedule be adapted in case of (1) late delivery of raw materials”; (2) delays in production runs; (3) changes in available stock caused by quality inspections; (4) rush orders; (5) break down of production lines;
- “Reject of the output of finished product at the end of the production line, how will the system adapt?”

These questions show a model-based and reductionist approach to the sales and operation planning process. The production system is provided a forecast from the central ERP system (just one value, no information about spread), processes this to an optimal production plan, and any deviation or disruption results in adjustments to the production plan. A fully deterministic production is assumed, as well as full and real time information about the actual situation on the shop floor.

This approach encounters a number of related fundamental objections: (1) production is not deterministic, as a result production results will always deviate from planning which leads to the next question: what constitutes a deviation? (2) product registration never fully coincides with production reality, the view on the computer screen is in the best case an abstraction of the shop floor reality (details are not available or omitted) and in other cases a distortion of reality (for example by enforcing classifications in search lists that have not been sufficiently thought through). And, imagine what this

approach would mean for the primary processes on the shop floor: a continuous flow of changes in production planning, which will go at the expense of effectiveness and efficiency.

6 RECAPITULATION AND CONCLUSIONS

The problem area is about how to combine agreed service levels with minimising costs. The Sales & Operations Planning literature clearly indicates the comprehensive character of the problem, different viewpoints have to be taken into account. In structuring the problem field the literature is less helpful, firstly because of the confusing use of terminology, secondly because of the time horizon of months and years, and thirdly because of the lack of attention for the day-to-day work and challenges for the people involved in planning. Software, as could be expected, does not help either. It offers toolboxes of statistical instruments, dashboards, and algorithms without much notion of how to apply which instrument in which situation. The fact that a planner is responsible for his decisions, and that the planner has to combine information from many sources in order to deal with variance, inaccuracy and conflict of norms is not discussed in the literature.

Essentially, the problem area is about two main control loops decoupled by a third intermediate control loop. The first main loop is about service levels, available stock and expected demand. The second main loop is about production output and efficient production. The intermediate loop is about stock control, firstly for decoupling variance of demand from smooth and efficient production processes, and secondly for dealing with all kinds of disruptions, deviations and inaccuracies in both the business processes itself and in available information about the business processes.

The approach is about finding solutions that do justice to both the structure of the problem (process logic, formal sign systems) and to the intricacies of the particular company that must organise its information in such a way that its competitive power and distinctive capabilities are enhanced (how it has found its own ways of dealing with the challenges, idiosyncratic characteristics, social sign systems). Human judgement and human communication must be combined with computing power.

In terms of the coordination mechanisms of Mintzberg we see a combination of standardisation (and formal sign systems) and mutual adjustment

(with social sign systems). Standardisation and rigid definitions of data help to organise information flows and to automate the processes of determining demand, setting stock levels, and setting production levels in standard situations. Dealing with conflicting norms in non-standard situations, however, is about mutual adjustment and human responsibilities. Squeezing out that little bit extra is about human creativity and problem solving.

To conclude with Kay: distinctive capabilities are about the characteristics that distinguishes the individual company from its competitors, and that can not be readily copied. The challenge is to find the right combination of formal and social sign systems that addresses the needs of actual planners in actual companies and that builds on the existing competitive power of the company. The high level talk of the literature nor the reductive approach of ERP software helps here much.

REFERENCES

- APICS, 2008. *APICS Dictionary*, APICS, Chicago IL.
- Boisot, M.H., 1998. *Knowledge Assets*, Oxford University Press, Oxford.
- Davis, R.A., 2016. *Demand Driven Inventory Optimization and Replenishment*, Wiley, Hoboken NJ.
- Dickersbach, J.T., 2009. *Supply Chain Management with SAP APO™*, Springer, Berlin.
- Grice, P., 1989. *Studies in the Ways of Words*, Harvard University Press, Cambridge.
- Hamilton, S., 2016. *Process manufacturing Using Microsoft Dynamics AX*, Scott Hamilton Press.
- Kay, J., 1998. *Foundations of Corporate Success*, Oxford University Press, Oxford.
- Mintzberg, H., 1979. *The Structuring of Organizations*, Prentice Hall, Englewood Cliffs NJ.
- Sheldon, D.H., 2006. *World Class Sales & Operations Planning*, J.Ross Publishing, Ft. Lauderdale FL
- Snapp, S., 2016. *Sales & Operations Planning in Software*, SCM Focus Press, Las Vegas NV.
- Starreveld, R.W., 1963. *Leer van de administratieve organisatie*, Samson, Alphen aan de Rijn.
- Suurmond, C.P., 2015. Information Systems and Sign Systems. *In Information and Knowledge Management in Complex Systems*. Springer, Berlin.

Assessing Business Processes by Checking Transaction Documents for Inconsistency Risks

Takafumi Komoto¹, Kokichi Futatsugi¹ and Nobukazu Yoshioka²

¹*Japan Advanced Institute Science and Technology, Nomi, Japan*

²*National Institute of Information Technology Grace Center, Tokyo, Japan*
{komoto, futatsugi}@jaist.ac.jp, nobukazu@nii.ac.jp

Keywords: Internal Control, Transaction Documents, Reliability, Inconsistency Risks, Checked Documents Matrix.

Abstract: Business processes can be assessed by checking transaction documents for inconsistency risks and can be classified into two categories. Inconsistency refers to a mismatch between items (product name, quantity, unit price, amount price, etc.) among transaction documents. For any process in the first category, the consistency of any pair of transaction documents in the process is checked, and there is no risk of inconsistency. For any process in the second category, the consistency of some pairs of transaction documents in the process cannot be checked, and there is a risk of inconsistency. This paper proposes a method for the assessment of risk inconsistencies. The assessment can be used to design and evaluate business processes for a company's internal control over financial reporting. A business process diagram and inconsistency risk detection algorithm for classifying business processes is provided.

1 INTRODUCTION

From the viewpoint of internal control, management has a responsibility to establish business processes that do not cause deficiencies over financial reporting. When deficiencies over financial reporting are pointed out by auditors, companies lose the reliability of their investors. (Shimizu, Nakamura, 2007); (Maruyama et al., 2008); (Sasano, 2006).

Certified Public Accountants (CPAs) examine the consistency among accounting transaction documents (slips, vouchers, etc.) related to transactions when performing an accounting audit. They check whether there is a mismatch between them and confirm the reliability of transactions. (Yamaura, 2002)

If such checks and confirmations performed by CPAs to posted transactions are incorporated into the business process, more reliable transactions may be realized. Company workers check between received slips and archived slips on the same transaction for consistencies in product name, quantity, unit price, and amount price in business processes. In other words, checking and confirming the consistency of transactions are already performed on-site.

However, these checks are independently

performed at each department of a company during the business process. Therefore, any inconsistencies among whole documents in transactions cannot be detected solely by checks performed in one department when such transactions pass through multiple departments.

For example, there are transaction documents "a", "b", and "c" in a transaction. When division "A" checks transaction documents "a" and "b", and division "B" checks transaction documents "b" and "c", inconsistencies in whole documents for the transaction are detected considering a transitive relation between "a" and "c" through "b". Conversely, when division "A" checks documents "a" and "b", and "B" only has document "c" any inconsistencies between them cannot be detected because there is no relation between "a", "b", and "c".

The detection of inconsistencies between transaction documents depends on what divisions check in transaction documents, i.e., the business process.

This paper proposes a method for assessment of risk inconsistencies. The assessment can be used to design and evaluate business processes for a company's internal control over financial reporting. A business process diagram and an inconsistency risk detection algorithm for classifying business

processes are provided.

The paper is organized as follows. The next section describes business process modeling using our business process diagram while Section 3 introduces an inconsistency risk detection algorithm for classifying business processes. Section 4 presents a case study. Section 5 discusses related work. Section 6 concludes this paper.

2 BUSINESS PROCESS DIAGRAM

A business process diagram is a diagram used to describe business processes of a company by listing business events and archived transaction documents and checked documents set. At first we will explain the elements and notations of the “business process diagram” using a simple example.

2.1 An Order-to-Delivery Process Diagram

This simple diagram (Figure 1) describes an “order-to-delivery process” in which a company orders goods from a vendor, and the vendor delivers the goods to the company. In this process, the company orders goods from the vendor with a purchase order document. When the vendor receives the purchase order document, it prepares the goods for shipping and delivers them with an invoice. The company receives the goods with the invoice and checks between the purchase order and the goods to ensure consistency with the invoice. The diagram in Figure 1 describes the “order-to-delivery process.”

In this diagram, **【vendor】** and **【company】** show entities. An “order” and a “deliver” are events in the transaction process. The events are indicated by arrows pointing from a sending entity toward a receiving entity of a transaction document. The events are run sequentially from top to bottom along a timeline of the entities.

The sides of an arrow are visualized by the following symbols to distinguish between a transmission and a reception.

- "●": start of the process, "▽": end of an event
- "△": start of an event, "▼": end of the process

On a side of the timeline between “▽” (end of an event) and “△” (start of a following event), work at the acceptance event can be described. (The work can be omitted.)

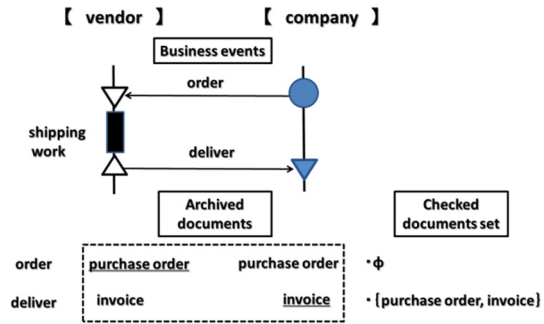


Figure 1: Order-to-Delivery Process Diagram.

In general, each transaction document is issued in accordance with a business event in the transaction. A “purchase order” and an “invoice” issued in this business process are described sequentially in the dashed frame indicating the archived transaction documents under the timeline of each entity. A line is drawn under a received document to distinguish it from a sent document. Business events, an “order” and a “deliver,” can be described in the side of the dashed frame to link to the transaction documents, a “purchase order” and an “invoice.”

In general, workers of a company also check between a received transaction document and archived transaction documents on the same transaction for consistencies in product name, quantity, unit price, and amount price in business processes. In the “order-to-delivery process diagram,” the state of transaction documents, whether checked or not by a division receiving a transaction document, is described.

At first, when **【vendor】** receives a purchase order, it does not keep any documents. Therefore, a checked documents set ϕ (empty set) is described. Next, when **【company】** receives an invoice, it keeps the purchase order. As **【company】** checks between the invoice and the purchase order, a checked documents set {purchase order, invoice} is described.

2.2 Elements and Notation of Business Process Diagram

As shown using the simple example, the business process diagram consists of the following elements.

- "Division": entity that performs the work in the process.
- "Timeline": time flowing from top to bottom.
- "Event": things needed to send and receive a transaction document from one division to

another division in a predetermined order.

- "Transaction document": documented work instructions and/or operating report in the business process.
- "Archived documents": transaction documents that the divisions sent and received.
- "Checked documents set": set of documents that the department is keeping (including a received document).

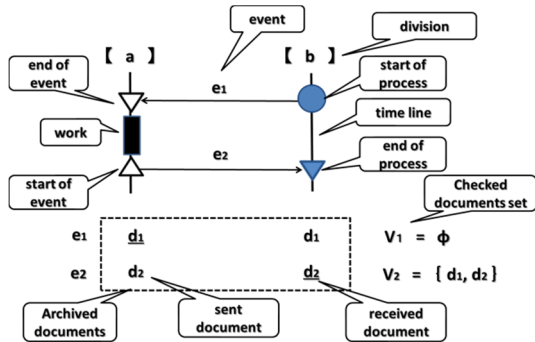


Figure 2: Business Process Diagram.

"Division", "Events", "Transaction document", "Archived documents", and "Checked documents set" are symbolized and defined as follows.

- Division $a, b \in \text{Div}$ (Div: the entire division)
- Event $e_n (a, b) \in E$ (E: the entire event): the n -th event to send and receive a document from division "a" to division "b".
- Event order $n \in N$ (N: natural number)
- Transaction document $d_n \in \text{Doc}$ (Doc: all documents): the document to send and receive in the event $e_n (a, b)$
- Archived documents $S_n (a)$: documents that division "a" sent and received until the event e_n
- Checked documents set V_n : set of the documents $S_n (a)$ that division "a" received the document d_n in the event e_n

The elements and notation of the business process diagram notation are shown in Figure 2.

2.3 Preconditions for Business Process Diagram

There are some preconditions for the business process diagram to represent practical standard business processes.

In a business process, when a person in charge in the division receives a transaction document, he/she works in accordance with business rules and issues a transaction document for reporting his/her task or

indicating a task of the next division. When he/she receives a transaction document from another division, and archive documents of the transaction are kept in this division, he/she can prevent an operational error by comparing the common items (product name, quantity, unit price, amount price, etc.) between the received document and archived documents.

Business process diagrams are used to detect inconsistency risks by examining mistakes or frauds. Accordingly, in the business process diagram it is assumed that transaction documents are not changed during storage and delivery. In other words, a sent document and a received document concerning the same event are regarded as the same.

It is also assumed that the event order of the business process is fixed. In general, business events in the company, in accordance with the principle of the separation of duty, are performed without being indicated by a transaction document. Therefore, in the business process diagram, the division not receiving a transaction document cannot send a transaction document except at the start of the event.

For example, in the purchase order process, the accounting division cannot pay for goods without receiving disbursement approval by the procurement division. In other words, each business event is carried out in the usual fixed order.

2.4 Example of Business Process Diagram at Risk for Inconsistency

Figure 3, which has a slightly modified business process diagram compared with Figure 1, [company] division of Figure 1 is divided into [purchase] division and [warehouse] division. The business event of receiving a report from [warehouse] division to [purchase] division is added.

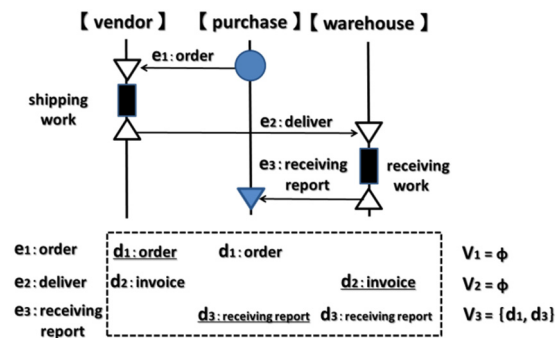


Figure 3: Business Process Diagram at Risk for Inconsistency.

Looking at the checked documents set V_i , received report d_3 and order d_1 is compared. However, invoice d_2 is not compared. Therefore, inconsistencies cannot be detected even if there is an error in the invoice. The business process diagram in Figure 3 is at risk for inconsistency of transaction documents.

3 INCONSISTENCY RISK DETECTION ALGORITHM

When a business process diagram is given, we provide an inconsistency risk detection algorithm that determines whether the business process has inconsistency risks among transaction documents.

The inconsistency risk detection algorithm is based on the equivalence relation of transaction documents. Transitive closure for the checked document matrix of the business process diagram is calculated using the Floyd-Warshall algorithm. (Cormen et al., 2009).

When the elements of the transitive closure matrix are all 1, no risk of inconsistency is decided. When the elements of the transitive closure matrix are 0, a risk of inconsistency is decided.

3.1 Documents Check and Equivalence Relation

“Documents check” compares common items of a received document to archived documents in the receiving division. Common items of transaction documents in the business process are product name and quantity, unit price, amount price, etc.

We determined that “documents check” serves as an equivalence relation as the result of the following analysis of “documents check.”

Document d_1 is naturally compared with itself (reflexivity law). When document d_1 is compared with document d_2 , document d_2 is compared with document d_1 (symmetric law). In addition, if document d_1 and document d_2 are compared, and document d_2 and document d_3 are compared, then document d_1 and d_3 have also been compared (transitive law).

Comparing reflexivity law and symmetry law is a convincing operation. For transitivity law, it has also been determined that a convincing operation can be assumed.

It should be noted that our discussion is based on the assumption of the sameness between the sent document and the received document, and the transitive law of “documents check”.

3.2 Inconsistency Risk Detection Algorithm

The state of the comparison with the entire set of transaction documents of business process diagram $Doc = \{d_1, \dots, d_n\}$ is represented by a matrix (Checked Documents Matrix).

Checked documents matrix $T(i, j)$ is set as 1 if document d_i and document d_j are compared. $T(i, j)$ is set as 0 if they are not compared.

Since the checked documents have an equivalence relation, the diagonal elements (i, i) are consistently 1 by reflexivity law, and (i, j) component and (j, i) component are equal by symmetric law.

We will explain the Checked Documents Matrix T using the following example. The entire set of documents of the matrix are $Doc = \{d_1, d_2, d_3\}$.

		d₁ d₂ d₃
Checked Documents Matrix T⁰	d₁	$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$
	d₂	
	d₃	

Checked Documents Matrix T^0 describes how document d_1 is compared with d_2 and d_3 , but document d_2 is not compared with d_3 .

However, document d_2 and d_1 are compared, and document d_1 and d_3 are compared in T^0 , so document d_2 and d_3 are also compared by transitive law.

At first glance, document d_2 and d_3 seemed not to be compared in T^0 . But matrix T^1 applying the transitivity law represents the true state of checked documents.

		d₁ d₂ d₃
Checked Documents Matrix T¹ of applying transitive law	d₁	$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$
	d₂	
	d₃	

As described above, continuing to apply the transitivity law for initial checked documents matrix T^0 , by calculating $T^1, T^2 \dots$, transitive closure T subsequently cannot be applied by the transitivity law any more. Transitive closure T represents the true state of checked documents.

Then, starting from the initial checked documents matrix T^0 , by applying the transitivity law, if the elements (i, j) of checked documents matrix T (transitive closure) are all 1, all the documents have been checked. Therefore, there is no risk of inconsistency in the business process. Conversely, if the elements (i, j) of transitive closure T include zero, no documents are checked with each other. Therefore, there is a risk of inconsistency in

the business process.

The inconsistency risk detection algorithm of the business process diagram is as follows.

<Inconsistency Risk Detection Algorithm>

- 1) Set the initial checked documents matrix T^0 .
 - All elements of T^0 are set to 0, and for Checked Documents Set V_i of the business process diagram, when V_i contains document d_i and d_j , (i, j) of T^0 is set to 1 for all i .
 - Diagonal elements of T^0 are set to 1. When the element (i, j) is 1, the symmetry element (j, i) is set to 1.
- 2) Calculate the transitive closure of checked documents matrix T^0 .
 - Calculate the T^n by applying the Floyd-Warshall algorithm (Cormen et al., 2009).

【Floyd-Warshall Algorithm (Cormen et al., 2009)】

The (i, j) element of the matrix T^k is t^{k}_{ij} .

for $k = 1$ to n

$T^k = a(t^{k}_{ij})$ is a new matrix

for $i = 1$ to n

for $j = 1$ to n

$t^{k}_{ij} = t^{k-1}_{ij} \vee (t^{k-1}_{ik} \wedge t^{k-1}_{kj})$

return T^n .

- 3) When the elements of the transitive closure T^n are all 1, there is no risk of inconsistency in the business process. When the elements of T^n are not all 1, there is some risk of inconsistency in the business process.

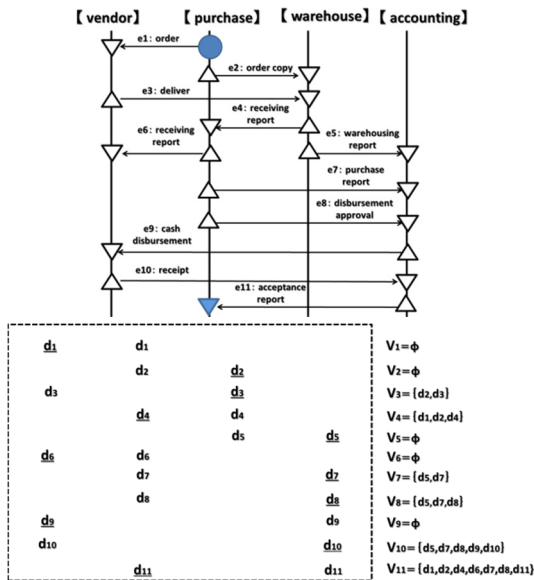


Figure 4: Standard Purchase Order Process Diagram.

4 CASE STUDY BY STANDARD PURCHASE ORDER PROCESS

The assessment of the standard purchase order process is performed in this case study. First, we make the business process diagram of the standard purchase order process (Figure 4) and extract the checked documents matrix from the checked documents sets V_i (for all i). Next, the inconsistency risk detection algorithm is applied for checked documents matrix T^0 , and the inconsistency risk of the process is judged.

4.1 Purchase Order Process Diagram and Inconsistency Risk Judgment

In the standard purchase order process, the purchase division orders goods from the vendor and notifies the warehouse division of the order. The vendor delivers the goods to the warehouse, and the warehouse receives them and sends the receiving report to the purchase division. The purchase division requests the accounting division for the payment in accordance with the invoice. The accounting division completes the disbursement and informs the purchase division about it to prevent duplicate payments. (Sasano, 2006); (Kaneko 2001)

This standard purchase order process diagram is shown in Figure 4.

The inconsistency risk detection algorithm is applied to the checked documents matrix T^0 , as shown in Figure 5. Since the elements of transitive closure matrix T^{11} are all 1, no risk of inconsistency in the standard purchase order process is determined.

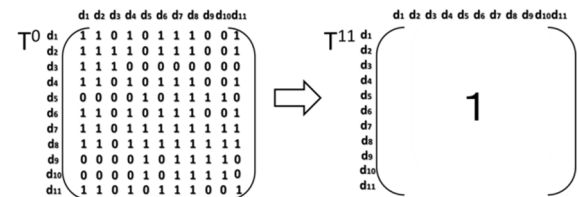


Figure 5: Transitive Disclosure Matrix T^{11} of Checked Documents Matrix T^0 .

5 RELATED WORK

We are currently unaware of any studies that model the business process by focusing on the documents generated in the business process and that assess the business process for inconsistency risks.

From the perspective of specific practical analysis of business rules and business processes,

the study described in this paper is considered to be unique.

Business process studies from the perspective of law compliance and standards are part of the field of business process compliance. These studies provide a framework for internal control in accordance with the Committee of Sponsoring Organizations of the Treadway Commission (COSO) and in accordance with health care privacy as established by the U.S. Health Insurance Portability and Accountability Act of 1996 (HIPPA) by analyzing the entire laws and standards. (Breux et al., 2006); (Siena et al., 2009) However, this paper does not provide a specific method that conforms to the standards established by COSO and HIPPA.

We are aware of a Resources, Events, and Agents (REA) study that analyzes and models financial accounting systems. In that study, all aspects of financial accounting are analyzed, but specific proposals for accounting audits are not provided. (McCarthy, 1982)

6 CONCLUSION

Comparison of received transaction documents with archived transaction documents by a person in charge of each division in a company is naturally performed to prevent any errors in the operation of each division. However, we cannot conclude that such a simple check in each division is enough to ensure consistency for the entire set of transaction documents in the business process, despite consistency in transaction documents belonging to individual divisions.

As indicated above, if the business process is properly designed, the consistency for the entire set of transaction documents is ensured. This operation approximately corresponds to auditing done by CPAs to confirm the existence of transactions.

This paper proposes a method of assessing business processes by checking transaction documents for inconsistency risks. This method consists of a "Business Process Diagram" and an "Inconsistency Risk Detection Algorithm."

Using the "Business Process Diagram" and the "Inconsistency Risk Detection Algorithm," business processes can be classified in two categories. For any process in the first category, the consistency of any pair of transaction documents in the process is checked, and there is no risk of inconsistency. For any process in the second category, the consistency of some pairs of transaction documents in the process cannot be checked, and there is a risk of

inconsistency.

When a business process is properly designed to meet the needs of the business process in the first category, inconsistency risks can be reduced.

We confirmed in the case study that the standard purchase order process established in the practices, due to the accumulation of experience over many years, is a business process in the first category.

This study aims to establish a high-quality method for inconsistency risk evaluation that can be incorporated into business rules and business processes by analyzing documents that are created on the basis of business rules and business processes. In this study, we modeled the business processes of transactions and assessed them for consistency risks. We will pursue logical verification by using CafeOBJ to refine our "Inconsistency Risk Detection Algorithm."

We will research a method to investigate mistakes and fraud in business processes in the future.

ACKNOWLEDGEMENTS

We thank Prof. Syuji Iida and Dr. Yasuhito Arimoto, Prof. Takao Okubo, Prof. Naoharu Kaiya, Mr. Motoharu Hirukawa, Ms. Junko Torimitsu for their valuable comments and feedback for our approach.

REFERENCES

- K. Shimizu, M. Nakamura, 2007: *Internal Control for IT Professionals*, Zeimukeiri Kyoukai (in Japanese).
- M. Maruyama, S. Kamei and T. Miki, 2008: *Readings from Internal Control Environment*, Shoeisha (in Japanese).
- M. Sasano, 2006: *Introduction and Practice of Internal Control*, Chuokeizaisha (in Japanese).
- A. Kaneko 2001: *Business Seminar Company Accounting Introduction*, Third Edition, Nihon Keizai Shimbun, Inc. (in Japanese).
- H. Yamaura, 2002: *Financial Auditing Theory*, second edition, Chuokeizaisha (2002) (in Japanese).
- T. Cormen, C. Leiserson, R. Rivest and C. Stein, 2009: *Introduction to Algorithms* [Volume 2], third edition, MIT Press.
- Travis D. Breux, Matthew W. Vail, and Annie I. Anton, 2006: *Towards Regulatory Compliance: Extracting Rights and Obligations to Align Requirements with Regulations*. RE 2006: 46-55.
- Alberto Siena, Anna Perini, Angelo Susi, and John Mylopoulos, 2009: *Towards a framework for law-compliant software requirements*. ICSE Companion 2009: 251-254.

McCarthy, E. W, 1982: *The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment*. *The Accounting Review*, (July 1982): pp. 554-578.

A Metamodel for Business Rules with Access Control

Lex Wedemeijer

Department of Management, Science and Technology, Open University, Valkenburgerweg 177, Heerlen, The Netherlands
Lex.Wedemeijer@gmail.com

Keywords: Metamodel, Declarative Business Rules, Access Control, Relation Algebra.

Abstract: Business rules outline the way of working with data in today's organizations. We present a metamodel to support and underpin the rule-oriented language to capture business rules that we developed earlier. Like the language, our metamodel is founded on Relation Algebra. The metamodel is compact, and enables the rule designer to record the business rules in their exact details, and to examine the data of the organization for rule violations. Even though such violations should be signalled to the appropriate stakeholders in the business, the access to such signals is subject to access controls, which constitute a special type of business rule. We account for this requirement in our metamodel, so that it captures regular business rules and access permissions alike, and enables to monitor them for violations. A prototype implementation demonstrated the feasibility of our approach.

1 INTRODUCTION

Rule engineering calls for a rule language that is understandable for the intended user community, yet precise enough for subsequent application development. The importance is well argued in the Business Rules Manifesto (2003).

In an earlier paper (Wedemeijer, 2015), we proposed a concise language to specify declarative business rules that consists of just 5 statements. We also proposed a provisional metamodel, but this has since been found to be inadequate. The lack of a proper metamodel hampered the further development of the rule-oriented language and design tools.

The objective of this paper is to present a metamodel to support our language for declarative and state-oriented business rules, including rules for access control, and to provide a solution to the long-standing problem how to reconcile access rules with regular business rules.

Binary Relation Algebra provides the rigorous foundation for our work. The variant we use can be classified as a Description Logic of SHIO type (Baader et al., 2008).

The paper is organized as follows. Section 2 points out some related work. Section 3 explains the major part of our metamodel in conjunction with four of the language statements. The fifth language statement concerns rule enforcement. This is analysed, and an alternative is proposed in section 4.

In section 5, we briefly discuss the rules that apply at the metamodel level. Violation of a metamodel-rule signifies an error in the design of the business model, or in a business rule formula. Section 6 presents the completed metamodel. Its integration with the rule language is underpinned by a prototype implementation. Section 7 concludes the paper.

2 RELATED WORK

Declarative business rules have been analysed and modelled in various contexts and from different viewpoints. Rule metamodels as well as access control models are well described in the literature. We mention some approaches that are related to our work on declarative business rules, without attempting to be exhaustive.

2.1 Metamodels for Business Rules

Object Constraint Language (OMG, 2012) is a language to describe constraints on classes defined in UML models. Its emphasis is on implementation, and we consider it too technical for our purpose. The SBVR current standard (OMG, 2015) describes a metamodel for business rules, but covers a much wider area than the declarative and state-oriented business rules that we focus on. Hence, the metamodel is overly complex and does not fit our purpose.

Ecore (Steinberg et al., 2008) is a metamodel for the Eclipse programming environment. Specified as an UML model, its aim is to describe models and to provide run-time support for datamodels in Eclipse. As it aims to also support operations, it is not truly state-oriented. Ecore too covers a much wider area than what we focus on, while a genuine notion to represent business rules is lacking.

The Semantic Web Rule Language (Horrocks et al, 2004) is a rule-based approach that employs Ecore as a basis. Semantic Web approaches use the Open-World Assumption whereas our work is based on the Closed-World Assumption (Ceravolo et al., 2007). This fundamental difference is most evident in the 'total' ruletype that requires inspecting all items of an object (class): the rule may be satisfied under Closed-World Assumption, but unknown under the Open-World Assumption.

RAP (Michels, 2015) is a language and metamodel geared to declarative business rules. However, it aims to provide learning support for students in a course for business rules, and the metamodel, tailored to this goal, is implementation dependent causing RAP to be deficient in its support of rule enforcement and access control.

In summary, few of the approaches discussed above support the state-oriented and declarative properties that we think are fundamental in business rules. With the possible exception of RAP, they are not suited to fit our compact rule language.

2.2 Models for Access Control

The contexts of access control and regular business rules are generally regarded as different: primary rules outline what must be done to create user value. Access control rules are secondary rules to outline what is, or what is not allowed in doing so.

Role-Based Access Control (ANSI, 2004) was defined in 2004 and is still being developed. It is a standard for access control that describes a strategy for granting permissions to view data and to perform editing operations. The model stipulates separation of concerns: users, sessions and roles on the one hand, and permissions for data-objects and operations on the other, the two contexts being linked by assigning Permissions.

Relation-Based Access Control (Zhang et al., 2010) is a variant that emphasizes this notion of linking-pin. Notice however that the operations and data-objects covered at the right-hand side of Core RBAC may concern data editing operations as performed by business users and thus subject to user assignments. Complications that may result from

this duplicity are ignored in RBAC.

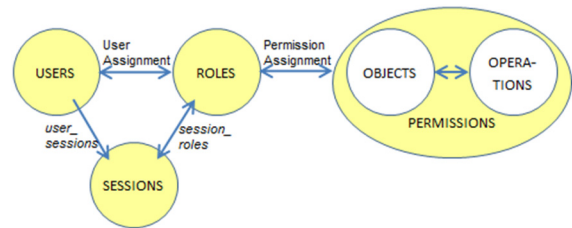


Figure 1: Core RBAC.

Access control rules can, and should be described as regular business rules (Liu et al., 2003). However, access control rules are separated from the primary business concepts, and potential matches or links are neglected.

Most access control approaches are preventive in nature, assuming that access attempts without prior permission will automatically fail. In business practice, this is not always how it works. It is often unclear how an approach copes with accesses that have actually occurred without a corresponding permission.

Access control rules may be regarded as rules in their own right. Still, to the best of our knowledge, no standard exists of a joint model combining rules about primary business data (concepts and relations) with secondary rules to control access to that data.

3 METAMODEL

Figure 2 depicts the major parts of the metamodel.

Rectangles represent concepts that in Relation Algebra have no attributes, unlike conventional Relational Algebra. A line with a name represents a relation. Our convention is to let arrowheads point from the domain to the range concept. Although less common, this is useful for designers in writing correct rule formulas. Dotted lines depict specialization-generalization relationships.

The three shaded areas correspond to the language statements as indicated.

Relations in the metamodel are univalent; the exception of [Tuple] *is_in* [Expression] is discussed later. A relation is total if the connecting line begins at the boundary of the domain concept. Otherwise, a line starting out from the interior, the relation is optional. This convention suggests that the relation is total for some specialization that is not depicted explicitly.

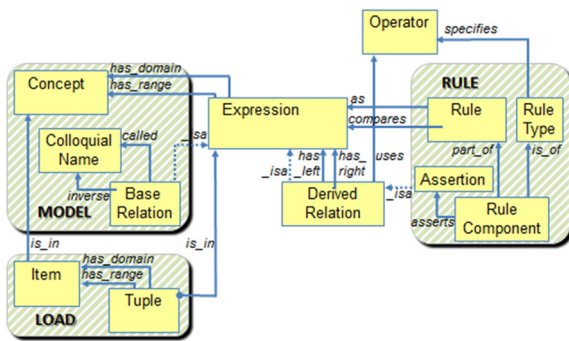


Figure 2: Partial metamodel for Business Rules.

3.1 The MODEL Statement

At the left are the concepts of Concept, BaseRelation and ColloquialName as specified in the MODEL statement. Together, the name, domain and range are unique for each BaseRelation. An inverse name may optionally be given, which must also be unique.

BaseRelation is our preferred name for relations that the designer explicitly specifies. This is to avoid confusion with expressions, constructed by way of Relation-Algebra operations.

Specialization/generalization is captured by way of *isa* (and inverse name *asi*) as a reserved ColloquialName, and there are several instances in the metamodel, depicted as dotted lines. A natural restriction for *isa* relations is that each one must be an injective function.

3.2 Regarding Expressions

Expression is the core concept in our metamodel. It captures binary relations created by way of Relation-Algebra operators. This is where the power of Binary Relation Algebra comes in, and no language statement is needed.

Once base relations are specified, other relations are derived, either by applying a unary operator to one expression, or by applying a binary operator to two expressions. The metamodel captures the two options by way of a left-hand argument and operator, which are always compulsory, and an optional right-hand argument.

The Operator concept is the set of operators such as inverse, negation, union, intersection, and composition, to be used in DerivedRelations. The actual list of available operators is implementation-dependent, and is easily expanded. For instance, a unary operator called 'total' may be implemented to derive all items in the domain concept that do not partake in a certain expression.

Apart from BaseRelations and DerivedRelations,

the metamodel also provides for IdentityRelations and nominals. A designer can refer to the Identity Relation for a concept in any language statement without having to define it first.

Nominal expressions are denoted as constant values, i.e. a single or multiple pairs. Although this resembles a population of tuples, nominals are fixed expressions that come without extensions and so cannot be edited.

3.3 The LOAD Statement

At the lower left in figure 2 are the Item and Tuple concepts and their associated relations that record the populations of Concepts and Expressions. A designer needs to specify tuples for BaseRelations only, because Relation Algebra will then determine all tuples for all Expressions according to the expressions' derivation formula. This applies at loadtime, but later at runtime as well.

By exception, the relation in the metamodel from Tuple to Expression is not univalent. An essential property of Relation Algebra is that one tuple can be member of multiple Expressions. By implication, one (set of) tuples can be loaded into more than one BaseRelation at once.

3.4 The RULE Statement

The Rule statement of our language is captured by Rule, RuleType, RuleComponent and Assertion, with corresponding relations.

One variant of the rule statement restricts a single expression, potentially by specifying several constraints at once. For example, stating that an expression MUST BE FUNCTION means that two rule-types apply, 'univalent' and 'total'. Applying objectification as a design pattern (Halpin, 2006), we reify the link between Rule and RuleType into the concept RuleComponent.

The other variant of rule statement compares an expression with another, using the MUST IMPLY comparison.

A minor language improvement is to introduce new comparators MUST INCLUDE, MUST EXCLUDE, and MUST EQUAL, variants that also compare two expressions. The designer can then pick the comparator that makes a rule easy to understand.

Each comparison gives rise to a single RuleComponent, except MUST EQUAL that combines the two variants MUST IMPLY and MUST INCLUDE.

Each RuleComponent is uniquely identified by its rule and type of rule. It comes with exactly one assertion, the derived relation containing all tuples

that violate the RuleComponent. Rules need not be unique, which is why a single assertion may link to more than one rule component.

3.5 The EXPLAIN Statement

The metamodel accounts for the EXPLAIN statement of our language by one concept, Text, related to the Concept, BaseRelation, and Rule concepts. We specify the relations as optional because textual explanations contribute to the users' understanding of the metamodel, but not to its formal consistency, correctness, or completeness. And although some form of organization or coherence in the texts may be desirable, we consider such nice-to-have documentary features beyond the scope of this paper.

3.6 regarding the ENFORCE Statement

Ideally, a business adheres to all of its rules at all times, and no violations occur. In the context of our metamodel, it means that all assertions ought to remain unpopulated. In a live business environment, rules may sometimes be violated, so that some assertions will record violations. If our purpose was only to develop a declarative, state-oriented metamodel for business rules, then the passive capability to record violations would suffice. However, a running business calls for more than that. The problems and alternatives for rule enforcement are discussed in the next section.

3.7 Discussion

The metamodel presented in figure 2 accounts for four of the five statements of our language. Consisting of only 12 concepts and 20 relations (*_isa* relations included), it is compact but enables the rule designer to model business concepts and relations, and to specify declarative business rules about them.

Rule violations are recorded as (sets of) tuples in assertions and are available for inspection by the business stakeholders in charge of remedying the violations.

4 RULE ENFORCEMENT AND ACCESS CONTROL

The metamodel will record rule violations, but does not enforce the rules. No mechanism is provided to guarantee rule adherence in the running database

environment. Indeed, it is implementation-dependent how violations will be determined or prevented. This section explains how we extended the metamodel to support rule enforcement, albeit not in full.

4.1 Analysis of ENFORCE Strategies

We now take a closer look at the three strategies to specify enforcements, proposed in our language as:

```
ENFORCE Rule AS Reject/Report/Resolve
```

ENFORCE AS REJECT is easy to handle. Whenever an attempt is made to change the current state of the database such that a new rule violation would emerge, then the change is rejected and the database state remains unchanged. It reflects the view that the population of the database after the change, with this violation, cannot possibly be true in the real world. In practice, this view may be valid for some rules, but certainly not for all.

The ENFORCE AS REPORT statement is an obligation to report violations to the stakeholder(s) in charge of monitoring rule compliance, and also a permission for the stakeholder(s) to read them. But the obligation to report specifies compulsory action, which is inconsistent with our purpose of declarative language and metamodel. Also, it tacitly assumes that the metamodel contains a stakeholder concept; this is even an explicit assumption in the RBAC standard discussed above.

In practice however, the stakeholder role is assumed by business people, and this role, or these business people, may already be captured as a concept. In such cases, adding a Stakeholder concept to the metamodel results in undesired redundancy. Each change in organization, stakeholdership, or monitoring responsibilities, would require an update in the metamodel. In our experience (Wedemeijer, 2002), such changes in the way of doing business are rather frequent, much more so than changes in rules, relations or concepts.

ENFORCE AS RESOLVE as the third strategy also poses serious difficulties. In our earlier paper we pointed out that this strategy is not declarative but imperative, and not state- but transaction-oriented in character. Moreover, it assumes that the stakeholder who performs the data edit to resolve the violation, has permission to access that data.

Enforcements are rules about rules. Enforcement rules in many practical business environments are phrased as imperative rules, whereas we are looking for declarative business rules. They constitute rules in their own right and so should be handled in much the same way as ordinary business rules. Because rule enforcement strategies implicate that rule

violations need to be accessed, we turn our attention to access control. Our aim is to capture and integrate the rules for data access into the metamodel as regular business rules.

4.2 Permission to Access

Rule enforcement is dependent upon access control. To report violations to some stakeholder, assumes that the stakeholder is permitted to read the violations, and that proper edit permits are granted for taking corrective actions.

In accordance with the RBAC standard, our access control rule is simply: "access to data requires permission to access that data". This applies to all expressions, not only to the assertions associated to some business rules.

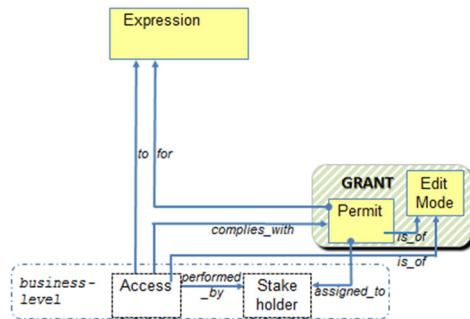


Figure 3: Permit and EditMode concept in the metamodel. Access and Stakeholder belong to the business level.

Figure 3 shows how the metamodel is expanded with a Permit and EditMode concept to cover the 'permission assignment' relation of RBAC. A permit allows a user to access the data recorded for that expression(s). Obviously, an expression cannot be validly accessed if no permits are granted for it, and so its contents remains hidden to the users.

EditMode is optional, to tailor permissions to various modes of data editing (add, change, delete). The default access mode is 'read' which does not refer to an editmode. Expressions in general cannot be edited, and editmodes should be made to apply only to BaseRelations or (the IdentityRelation of) Concepts. Evidently, an edit permit automatically implies read permission for that expression.

4.3 Access Control Rules

Access control requires keeping track of all data access. But keeping score of all data accesses comes down to record the online transaction processing as performed in the running business database environment: we do not suggest to duplicate this

within the context of our rule metamodel which would become very volatile.

Instead, we assume that some concept of Access is defined at the business model level (figure 3). In the running database, each access should be recorded: to which expression, and compliant to which permit. These relations, transgressing the business/metamodel boundary, support the decision-making process in the operational database to allow, or deny access.

Next, we also assume two relations: [Access] *to* [Expression] and [Access] *complies_with* [Permit]. Both relations will be functions (univalent and total), if the Access concept is defined suitably.

Of course, other and more complicated concepts and rules for data access may exist in more realistic business environments. And there is more to it. Access to an expression requires that the access comes under a permit for that exact expression:

```
RULE 123-read-control AS
  [Access] to [Expression]
MUST IMPLY
  [Access] complies_with [Permit]
  composition
  [Permit] for [Expression]
```

The statement follows the language conventions set out in our earlier paper (Wedemeijer, 2015). The rule-identifier, 123-read-control, is arbitrary.

If access involves editing, then an extra restriction is that the permit allows the correct editmode:

```
RULE 456-edit-control AS
  [Access] is_of [EditMode]
MUST IMPLY
  [Access] complies_with [Permit]
  composition
  [Permit] is_of [EditMode]
```

4.4 Access Control for Stakeholders

A next extension is to account for stakeholders who actually perform the data accesses. In ordinary business models, stakeholders go under a wide variety of names: users, employees, departments, or whatever. Most approaches for access control, and the RBAC standard is no exception, capture such concepts in a separate business model. Our approach leaves it to the business designer to specify in the business datamodel who the permissions are granted to, and to specify the volatile relations of these concepts with Permit. This allows the designer to merge this important aspect into the overall business model, or to employ a secondary model.

To outline how this works out in practice, we

assume for the sake of this paper that a single Stakeholder concept is identified in the business datamodel as in figure 3. We also assume the [Access] *performed-by* [Stakeholder] relation which is univalent and total, and relation [Permit] *assigned-to* [Stakeholder] which may be many-to-many. The rule that access requires permission now comes down to:

```
RULE 789-access-control AS
  [Access] performed-by [Stakeholder]
MUST IMPLY
  [Access] complies_with [Permit]
  composition
  [Permit] assigned-to [Stakeholder]
```

Notice how these rules apply at the level of the business model, and violations will appear in a regular Assertion. From a business point of view, the only peculiarity is that the Expression and Permit concept and contents are fixed at the metamodel level, and cannot be edited on the fly.

Combined, the rules 123, 456 and 789 support all three enforcement strategies, as follows. If no permit was issued for a rule, then viewing violations is impossible so no violations should emerge, which comes down to the "reject" strategy. If some permit was issued, then stakeholders with a proper permit can obtain a report of the violations, which is the "report" strategy. If an edit permit was issued, then the stakeholder may proceed to add, change or delete a violating tuple, executing the "resolve" strategy.

4.5 GRANT Statement Replaces ENFORCE

As the ENFORCE language statement is inadequate, we now propose as alternative the GRANT statement. It records permit-identifiers, and relates them to one or more expressions in the metamodel. One variant issues permits for reading, the other for editing:

```
GRANT Permit FOR Expression(s)/Rule(s)
GRANT Permit FOR BaseRelation(s)
  IN EditMode
```

For the convenience of the business designer, a single permit may be granted for several expressions and (assertions corresponding to) rules at once, or one edit permit for multiple base relations. Once the permits are established in the metamodel, they can be assigned to any roles or users as defined in the business model, at load-time or at runtime of the operational database.

The statement specifies access control for all expressions, not only for rule assertions. Rather, an

enforcement strategy can be inferred from the permissions granted for assertions.

If no permit is granted for an assertion, then violations cannot be viewed and so should not exist. Hence, the rule ought to never be violated, and the reject strategy applies. A read permit granted for an assertion means that rule violations are possible, and some stakeholders are probably able to view them: this is the report strategy. If an edit permit is granted, then the resolve-strategy applies.

4.6 Discussion

Our way of granting access permissions has great advantage. The point is that a distinct assertion will record the violations of the access-control rule. A distinct read permit is required to inspect access violations, which must be defined in the metamodel and then assigned to stakeholders in the business.

For example, access to some data may be performed by a stakeholder without the proper permission, resulting in a violation of rule 789-access-control. One possible way to resolve the violation is by having the permission assigned to the stakeholder belatedly. This is a great feature for data auditors, and rarely supported in other approaches for Access Control.

The granting of permissions resembles the RBAC standard that also envisions separation of concerns, but ignores the potential overlap of (user) roles with objects in the business model. Our approach allows such overlap, and puts the business designer in charge of avoiding possible duplicity.

The main advantage is that access control is now relegated to the level of business model, where permits can be assigned, changed and withdrawn at any time, without affecting the metamodel. Thus, access rule violations can be handled in the same way as violations of ordinary business rules.

One may argue that a mere permission to read violations, does not ensure that a stakeholder will actually do so. But this is equally true for the ENFORCE AS REPORT statement: there is no recordkeeping of violations that have or have not been viewed. Would a rule be imposed that each violation must be viewed, then each emerging violation constitutes a violation of this rule. This is inferior design: the simultaneous emergence of two violations is a clear update-anomaly.

5 METAMODEL-RULES

In any model of data, rules apply to ensure correct

ness and consistency within the modelled context. A rule violation signifies that some data is flawed, and should be remedied by the stakeholder. In our metamodel, violation of a metamodel-rule signifies that some feature of the business model is flawed, and should be remedied by the designer.

5.1 Cardinality

Univalence applies to almost all relations in the metamodel. The exception in relation [Tuple] *is_in* [Expression] was already pointed out. A second exception is the [Permit] *for* [Expression] relation. Notice that arbitrary cardinalities may apply to relations involving a Stakeholder, but that concept is part of the business model, not the metamodel.

Many, but not all relations are total. An important optional relation concerns permits that not always involve an editmode. Other relations that are not total are: the inverse name of BaseRelation; the right-hand side in a derivation formula for a derived relation; and the relations for explanatory texts. The three *isa* relations in our metamodel combine univalence, totality, and injectivity.

5.2 Compound Rules

Compound rules involve more than just one relation.

In our metamodel, an example is the short loop between Rule and Expression: the *compares* relation must always refer to a different expression than the *as* relation, so as to avoid trivial and contradictory formulas.

A more interesting example is the requirement of referential integrity. Not only must the items in a tuple be on record for some concept, they must exist in the correct concept. This boils down to two compound rules, one about domains and the other about ranges:

```

RULE 0-referential-integrity-domain AS
  [Tuple] is_in [Expression]
    composition
  [Expression] has_domain [Concept]
MUST EQUAL
  [Tuple] has_domain [Item]
    composition
  [Tuple] is_in [Concept]
    
```

Another consistency rule for a metamodel loop concerns [Rule] *compares* [Expression]. Whenever a rule uses *compares*, then must its corresponding assertion, as a DerivedRelation, have a *has_right* [Expression] clause that uses that exact same expression.

A more intricate rule concerns the Operator concept. The operator used in the assertion for a rule component, is fully determined by that components' RuleType. For example, if a rule component expresses that some expression is 'total', then the corresponding assertion must use the specific 'total' unary operator.

Apart from compound rules that apply to loops in the model, other compound rules can be pointed out. For instance, a compound uniqueness (identifying) rule applies for BaseRelations: each is uniquely identified by its ColloquialName, plus its domain and range concept. Likewise, each Tuple is uniquely identified by the combination of its domain item, and its range item.

Still another kind of rule concerns *isa* hierarchies: [Assertion] *isa* [DerivedRelation], composed with [DerivedRelation] *isa* [Expression], must coincide with the [Assertion] *isa* [Expression] relation. The latter relation is not depicted, but it does exist as a proper specialization/generalization.

6 THE INTEGRATED VIEW

Figure 4 depicts the compact metamodel to support our business rules language. The ENFORCE statement of our earlier language version was found to be imperative, and too volatile. Focusing on access control only, we defined a GRANT statement for access permits, to regain the declarative and state-oriented character of the language and to reduce volatility.

6.1 Language and Metamodel

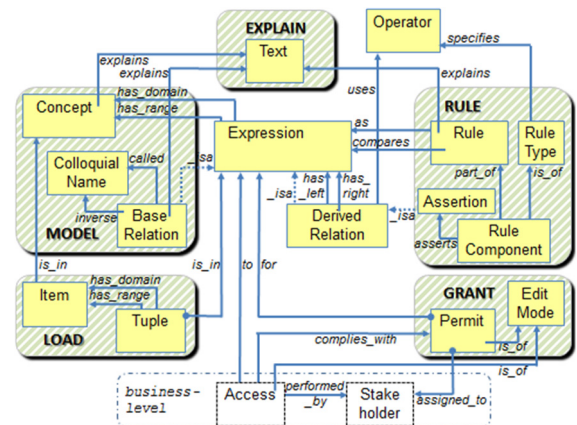


Figure 4: Full Metamodel for Business Rules.

The current language supports five statements: MODEL, RULE, EXPLAIN, LOAD and GRANT. Jointly, they enable a business designer to set up a model, to

specify rules, to provide explanations, to load initial data, and to determine access control. The supporting metamodel is expressive yet compact, as twelve concepts suffice to capture a design.

Remind that Expression, DerivedRelation and Operator need not be specified by the designer as they come for free by virtue of Relation Algebra. And although a Stakeholder and Access concept are depicted, these concepts and relations are not part of the metamodel. Permit assignation to stakeholders, and actual access of data, should be recorded at the business level. In our experience, this will considerably reduce the number of changes in the metamodel.

6.2 Demonstration

The metamodel and rules constitute a business model just like any other, perhaps with a somewhat peculiar context. Hence, it can be expressed in our rule-oriented language, and captured as a regular datamodel in the metamodel itself, in a reflective fashion.

Feasibility of our approach is demonstrated in this way, by implementing the metamodel and its complete set of rules in a prototyping rule-based engineering environment. The result is available for download at wiki.tarski.nl/index.php/Research_hub.

7 CONCLUSION

The metamodel defines the information structure underlying our language for declarative business rules, and also covers rules for access control.

Binary Relation Algebra is used as theoretical fundament for exact rule specifications. This formalism however does not support numerical, temporal, or spatial capabilities for rules. Nor is the metamodel designed for performance or scalability, and no efficient algorithm is proposed to determine rule violations. Deployment will call for a distinct development step to transform the business model to a proper database schema that takes requirements into account such as performance, data distribution, federation across hardware platforms, security, interoperability etc.

The metamodel ensures separation of concerns, so that business users can add, edit and delete the operational data, provided that a proper permit was assigned to them. If not, the violation of the access control rule is captured in a rule assertion, just like any other business rule violation.

A salient point of the metamodel is that dependence on role and permit assignments is minimal.

Thus, it provides a stable environment to capture and describe business rules. Volatility due to everyday changes in organizations is relegated from the metamodel to the level of the business model.

Our approach handles the primary business rules and the rules for access control in exactly the same way, an elusive goal in business rules engineering attested to in the Business Rules Manifesto. The metamodel for business rules with access control presented in this paper indicates how this goal may be reached.

REFERENCES

- ANSI, 2004. *Role Based Access Control*. American National Standard for Information Technology INCITS 359. available at csrc.nist.gov/groups/SNS/rbac/
- Baader, Horrocks, Sattler, 2008. *Description logics*. In: van Harmelen, Lifschitz, Porter. (eds.). Elsevier Handbook of Knowledge Representation, pp 135–179
- Business Rules Manifesto, 2003. *Version 2.0*. available at www.businessrulesgroup.org/brmanifesto.htm
- Ceravolo, Fugazza, Leida, 2007. *Modeling Semantics of Business Rules* Inaugural IEEE-IES Digital Eco-Systems and Technologies Conference, pp. 171-176
- Halpin, 2006. *Objectification of relationships*. Advanced Topics in Database Research. Vol 5 no 5 pp 106-123
- Horrocks, Patel-Schneider, Boley, Tabet, Grosz, Dean, 2004. *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. Nat.Research Council of Canada. available at www.w3.org/Submission/SWRL/
- Liu, Sun, Barjis, Dietz, 2003. *Modelling dynamic behavior of business organisations*. Knowledge-Based Systems, pp. 101-111
- Michels, 2015. *Development Environment for Rule-based Prototyping*. Dissertation. available at portal.ou.nl/documents/114964/31994420/Development_Environment_for_Rule-based_Prototyping.pdf
- Object Management Group, 2012. *Object Constraint Language*. available at www.omg.org/spec/
- Object Management Group, 2015. *SBVR: Semantics of Business Vocabulary and Business Rules*, Version 1.3. available at www.omg.org/spec/
- Steinberg, Budinsky, Paternostro, Merks, 2008. *Eclipse Modeling Framework, 2nd Ed*. Addison-Wesley
- Wedemeijer, 2002. *Exploring Conceptual Schema Evolution*. Dissertation. ISBN 90-5681-142-8. available at repository.tudelft.nl/
- Wedemeijer, 2015. *A Language to Specify Declarative Business Rules*. Springer Lecture Notes in Business Information Processing. Vol 220, pp 82-101
- Zhang, Giunchiglia, Crispo, Song, 2010. *Relation-Based Access Control: An Access Control Model for Context-Aware Computing Environment*. Wireless Personal Communications. Vol 55, pp 5-17

Business Models & Business Cases for Point-Of-Care Testing

A. J. Staring, L. O. Meertens and N. Sikkel

University of Twente, PO Box 217, Enschede, The Netherlands
a.j.satring@alumnus.utwente.nl, {l.o.meertens, k.sikkel}@utwente.nl

Keywords: Business Modelling, Modelling Method, Business Model, Business Case, Healthcare.

Abstract: Point-Of-Care Testing (POCT) enables clinical tests at or near the patient, with test results that are available instantly or in a very short time frame, to assist caregivers with immediate diagnosis and/or clinical intervention. The goal of POCT is to provide accurate, reliable, fast, and cost-effective information about patient condition. POCT can be part of the solution to the rising healthcare and welfare costs without any loss of healthcare quality. In this research, business models are used to create business cases in order to assess the viability of POCT. Two methods to create business models were designed by tailoring and extending them from an existing method. It was found that the method used has impact on the resulting business case. POCT was assessed to be viable in all business cases created for the specific case study used.

1 INTRODUCTION: POINT-OF-CARE-TESTING

Healthcare and welfare costs are rising, in the Netherlands, as well as other developed countries. Solutions need to be found to keep these costs within reasonable limits (Busse, 2001; Hagist and Kotlikoff, 2006), without loss of healthcare quality (Davis and Erixon, 2008). Point-Of-Care Testing (POCT) can be part of the solution. The goal of POCT is to provide accurate, reliable, fast, and cost-effective information about patient condition (St-Louis, 2000). Ehrmeyer and Laessig (2007) define POCT as “patient specimens assayed at or near the patient with the assumption that test results will be available instantly or in a very short time frame to assist caregivers with immediate diagnosis and/or clinical intervention”.

Already several forms of POCT exist, such as glucose testing and urine dipsticks (Altieri and Camarca, 2001). These used to be laboratory tests exclusively, but have evolved to focus solely on measuring the most critical parameters of the designed test. By focusing on only a few parameters, the test becomes more specific, faster, and the devices smaller (Dondelinger, 2009).

Although many advantages of POCT have been proven, such as fast diagnosis (Middendorf, 2010), error reduction (Drenck, 2001), and reducing the time patients stay in hospitals (Kiwa Carity, 2014), it has been pointed out that POCT may not be a cost-beneficial development always (Lehmann, 2001).

2 RESEARCH GOAL AND METHODOLOGY

This research aims to provide insight in the viability of POCT as part of the solution to the rising healthcare and welfare costs. We focus on a single case study in Dutch health care. For this case study, we create and evaluate several business cases. Business cases are a generally accepted way of assessing costs and benefits, as they provide “a justification for a proposed project or undertaking on the basis of its expected commercial benefit” (Oxford University Press, n.d.). To avoid creating arbitrary business cases from scratch, we apply the Business Modelling Method (BMM) (Meertens et al., 2012) and subsequently the Business Model to Business Case method (BM2BC) (Meertens et al., 2013).

In the process, we aim to validate the BMM’s use to create business models in a structured and repeatable way. In addition, extension of the BMM towards business cases may improve its value and applicability.

The process is as follows. First, we develop several derivatives of the BMM, so that we have methods to create business models that are tailored for healthcare and POCT specifically (section 3). Then, we apply these methods to create business models for our case study (section 4). Finally, we create business cases based on the business models and analyse them to assess their viability (section 5).

3 TAILORING THE BUSINESS MODELLING METHOD

The basic BMM consists of four steps to create a single business model in a structured and repeatable way (Meertens et al., 2012). As such, it aims to fill the research gap “Design tools” as defined by Pateli and Giaglis (2004) and Vermolen (2010).

The four basic steps are: 1. Identify roles, 2. Recognize relations, 3. Specify activities, and 4. Quantify model. The result of each step serves as input for the next step. The resulting business model is suitable for analysis of the current situation. It can also form the basis for further predictions, such as business cases, scenarios, and alternative innovations.

When using business modelling to evaluate new or alternate business ideas, two additional steps should be followed: 5. Design alternatives, and 6. Analyse alternatives. These steps are not strictly part of the method to create business models. The first four steps can even be reused when designing the alternatives. In addition, several other steps could follow, such as implementing the chosen business model alternative. Section 3.2 provides details of each step.

While for each of the steps example methods are given, none of these is enforced. This allows tailoring the BMM to the demands of the specific situation.

3.1 Extending the BMM

One extension step for the BMM is considered in this research. It splits step 4 (quantify model) of the basic BMM into two separate parts for costs and benefits. This makes it more suitable for business case analysis.

While step 4 of BMM aims to quantify the model, previous work only focusses on the costs of the model (Meertens et al., 2012). As a business that only spends money will not last long, revenue needs to be generated. Quantifying using only cost accounting methods is not sufficient.

Pricing methods can be used if to model “To be”-business models. To model the “As is”-business model, numbers from accounting systems and (annual) reports can be taken. To model the “To be”-business model pricing methods need to be used to determine the price.

Three approaches can be taken to set a price: cost-based, competition-based, and customer-based (Peter et al., 1991). Each of these approaches has its own advantages and disadvantages, as well as specializations.

3.2 Tailoring: Two Derivations of the BMM

While the previous sections outline the global process of the BMM, in this section we will provide two derivatives of it. These derivatives, BMMa and BMMb, are tailored by choosing appropriate methods for each of the steps. As Van Dijk (2015) identified roles and recognized relations for the same case study, steps 1 and 2 follow the methods he used. Only from step 3 onwards do the two derivatives differ from each other.

3.2.1 Step 1: Identify Roles

The first step focusses on identifying the relevant parties (i.e. roles) involved in a business model. Originally, the BMM suggests stakeholder analysis for this step.

In the preliminary research, Van Dijk (2015) chose to use the three-stage stakeholder analysis method presented by Pouloudi (1998). This method was designed for healthcare settings, such as our case study. For these reasons, we include this method in step 1 for both derivatives.

3.2.2 Step 2: Recognise Relations

The second step aims to recognise and characterise the relationships among the roles identified in the first step. Step 2 prepares for step 3, and follows naturally from the first step. The result can be as simple as a role-relationship matrix (Meertens et al., 2012). While a stakeholder analysis often follows a hub-and-spoke pattern, the second step of the BMM forces to specify and rethink all possible relations between the roles. The relations consist of interaction between two roles, with some form of value exchange. This is in line with Gordijn and Akkermans (2001) who state that all roles in a business model can capture value from the business model.

From this perspective, the proposed technique for this step, e3-value modelling, is a valid one. The e3-value model models the economic-value exchanges between actors (Andersson et al., 2009; Kartseva et al., 2006). This economic-value exchange can be tangible as well as intangible (Allee, 2008; Andersson et al., 2009). Van Dijk (2015) used e3value to visualise value exchanges with and without MobiHealth for the case study of this research. For these reasons, we include e3value in step 2 for both derivatives.

3.2.3 Step 3: Specify Activities

The third step specifies the activities needed for the roles to maintain their relationships and value exchanges. Every role-relationship, as recognized in the previous step, consists of at least one interaction between two roles, requiring activities by both roles. The activities reveal what should happen for the business to function properly.

Originally, the BMM proposes techniques from business process management to create the intended output. For tailoring, we follow this guideline and limit the difference between BMMa and BMMb to the choice of Business Process Modelling Language (BPML).

For BMMa, we choose Business Process Modelling and Notation (BPMN) (OMG, 2006). BPMN models consist of diagrams for both business users and developers. BPMN aims to simplify the understanding of business activity flows and processes. BPMN is chosen as it is the de facto standard for business process modelling.

For BMMb, we choose Event-Driven Process Chain (EPC) (Scheer and Schneider, 1992). EPC was developed for modelling business processes with the goal to be easily understood and used by business people. EPC is chosen, as it aims to be understood by business people, similar to the business models and cases for this case.

3.2.4 Step 4: Quantify Model

The fourth step turns the qualitative model into a quantitative model by obtaining numbers on cost and volume of the activities, which the previous step specified. This step helps to see what is happening in more detail, and allows for objective comparison between business models (Meertens et al., 2012). Numbers on cost and volume of activities are needed to completely overview the costs captured by the business model.

The origin of numbers depends on whether the “As is”-business model or “future”-business model is being modelled. To model the “As is”-business model, numbers from accounting systems and (annual) reports can be taken. To model the “To be”-business model, cost accounting methods need to be used to estimate the costs.

For an organisation to assign costs, several systems are available, both traditional systems and more refined systems, such as Activity-Based Costing (ABC) (Drury, 2008). Two main types of cost accounting methods exist: absorption costing, and variable costing. Their calculations differ mainly on the way they allocate overhead. Therefore, their

results mainly differ when sales and production do not match (e.g. seasonal sales with production in the rest of the year).

ABC in general is a type of absorption costing, but is argued to have its flaws (Anderson and Kaplan, 2003). Time-driven Activity-Based Costing (TD-ABC) is an improvement of basic ABC by requiring estimates of only two parameters: unit cost of supplying capacity, and time required to perform a transaction or an activity.

For BMMa, we choose the modern TD-ABC method, as its parameters best match the available input for the case study.

For BMMb, we choose variable costing as described by Drury (2008). Variable costing is considered less complex, and therefore better matches the choice for understandability by business people. As healthcare in general has service-based organizations, sales and production occur at the same moment, and therefore, the results should be the same as for absorption costing.

3.2.5 Extension Step: Pricing Method

This extension adds to the previous step by focusing on quantifying the revenues instead of the costs. As we want to design not only “As is” models, but also “To be” models, a pricing method is needed to determine the future revenue.

The most common used pricing method is the cost-based pricing method called cost-plus pricing (Drury, 2008; Gregson, 2012), also known as mark-up pricing. A limitation of cost-plus pricing is that demand is ignored. The price is set by adding a mark-up to the cost, and this may bear no relationship to the price-demand relationship. It assumes that prices should depend solely on costs (Drury, 2008). An other frequently used pricing method is value-based pricing (Gregson, 2012). The difference between value-based pricing and cost-plus pricing is “pricing down from value versus pricing up from cost”. Value-based pricing is a customer-based pricing method as the added value for the customer is translated into a price. Value-based pricing is setting a price that accurately reflects customers’ perception of value and proposes a process to do so (Bernstein and Macias, 2002).

For BMMa, we choose cost-plus pricing, as it is most commonly used method.

For BMMb, we choose value-based pricing, as this allows to confidently set the right price, as well as making a realistic prediction of revenue and profitability, which are two objectives of the process.

4 CASE STUDY: BUSINESS MODELS FOR POCT BY MOBIHEALTH

The eHealth technology provider MobiHealth is the main actor for the case study of this research. MobiHealth is a Dutch company that was founded in 2007. The company's roots lie in the European projects MobiHealth and HealthService24. In these projects, a prototype for mobile telemonitoring was designed, tested, and clinically validated in several European countries.

As this case study focusses on POCT, we will first describe the setting in which this takes place. However, the main part of this section applies the derivatives of the BMM to the case. The "As is"-business models reflect the situation without automated processing of POCT by MobiHealth. In the "To be"-business models MobiHealth takes part and performs offers different services. The "As is" and "To be"-business models reflect the same situations using two methods, BMMa and BMMb.

4.1 Case Study Setting: POCT at a General Practitioner in the Netherlands

When a sample needs to be taken for a test, this can either be done at the general practice (by the assistant or the general practitioner) or at the laboratory. Whether or not the sample is taken at the laboratory or at the general practice, it needs to be processed by the laboratory. If the sample is taken at the general practice, the sample needs to be transported by courier to the laboratory. Depending on the location of the general practice and the production volume (i.e. number of samples taken), the courier will visit the general practice at least once a day. When the sample is transported to the laboratory, the laboratory performs the necessary test(s) on it using specialised equipment. The test results are printed; a so-called printout. The printouts are entered by hand into the laboratory information system (LIS) and double-checked by a second pair of eyes. A third party transports the test results entered into the LIS to the system of the general practitioner (HIS; Huisarts informatie systeem). When the test results are entered into the HIS, the general practitioner is able to check the test results of their patients.

Some general practitioners, not all, have POCT equipment in their practice. This allows them to not only take test samples, but also test it using the POCT device. The result is shown on the display of the

device and allows the general practitioner to act upon the result immediately. The test results displayed on the POCT device need to be transcribed on a form or can be printed from the device directly. These forms and printouts are transported by the courier to the laboratory to be entered into the LIS. Sometimes, the data is sent by email to the laboratory either by scanning it in or the assistant typing the email. The processing of test results can take up to a couple of days. Performing tests on test samples can, depending on the capacity of the equipment and volume, also take a couple of days. Although the general practitioner has the test results immediately available using the POCT device, still the processing takes a couple of days.

MobiHealth has developed a service to process the test results of the POCT device automatically. They do this by adding hardware to the POCT device to read the test results and send it to the computer of the general practitioner. The results are then sent directly into the LIS. Using the existing integration between the LIS and HIS, the test results are entered into the HIS. This eliminates the time needed for the courier to take the results to the laboratory and the time it takes to enter the data into the LIS.

The laboratory supplies reagents to the general practitioner. These reagents can be used in the POCT device, but also to take samples or perform other medical operations. Only when a general practitioner submits a test sample or printout (by either courier, email, or using the service of MobiHealth) the laboratory is able to claim a reimbursement at the health insurer. The reimbursement is a fixed negotiated amount per test (amount depends on the type of test). Because of this, it is very important for the laboratory that performed tests are submitted by the general practitioner.

4.2 Business Model Foundation: Identified Roles and Relations

Van Dijk (2015) did the first two steps of the BMM for this case in previous work. Therefore, his work is a solid foundation to build upon in the following sections.

The first step of BMM, identifies roles taking part in the business model. The roles identified by Van Dijk (2015) is an extensive list. Not all roles can be found in the e3value model, as several are grouped. Groups of roles show a relationship more clearly than the individual roles themselves. Also it simplifies the situation and increase transparency and understanding. The listed roles match the roles in the extended e3value model (Staring, 2015). The roles can be found in Table 1.

Table 1: Identified roles, including descriptions.

Courier	transports reagents from the laboratory to the general practice and test samples and print outs from the general practice to the laboratory.
General practitioner	works in a general practice; multiple general practitioners can work at a single general practice.
Health insurance company	reimburse general practitioners and laboratories; issue health insurance policies to (potential) patients.
Patient	person in need of medical attention..
MobiHealth	provides integration service between POCT device and laboratory.
Laboratory-HIS integration company	provides one-way integration service between laboratory information system (LIS) and the information system of the general practitioner (HIS).
Laboratory	responsible for performing tests on test samples.
POCT device supplier	manufacturer/seller of POCT devices

The second step of BMM requires determining the relationships among the roles. Figure 1 shows the relationships in a simplified e3value model for a situation with courier, “As is”-model. Several of the roles and relationships that were in the extended model have been left out, as they are not influenced by the “To be”-business models.

4.3 As Is Models

4.3.1 Specify Activities

Four processes are identified from Figure 1. These are the activities for the business model, as shown in Table 2. For each of the process, business process diagrams have been elaborated by Staring (2015).

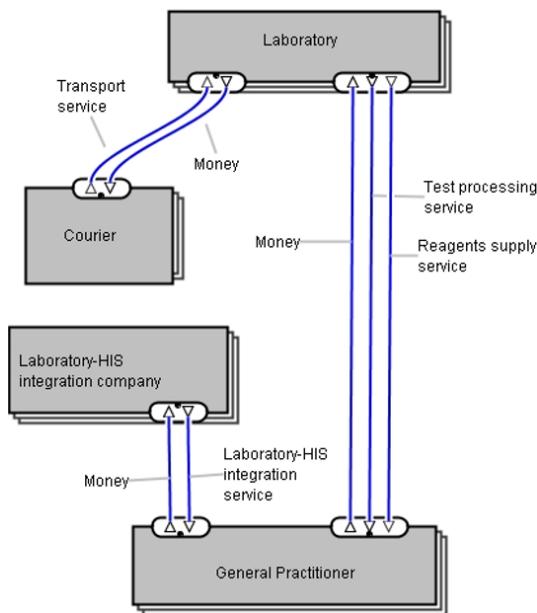


Figure 1: Simplified "As Is" e3value model.

The reagents supply process relates the reagents

supply service and the test processing process. Both services are provided by the laboratory and are used by the general practitioner. The test processing process uses two other processes: the transport process (related to the transport service provided by the courier) and the laboratory-HIS integration process (related to the laboratory-HIS integration service provided by the laboratory-HIS integration company). The transport process is used by the laboratory to pick up the test samples from the general practitioner. The laboratory initializes the laboratory-HIS integration process, but it is of value to the general practitioner as this enables test results to flow back into their systems (HIS).

4.3.2 Quantify Model

The specified activities in the previous step are quantified to determine the costs. Table 2 shows the results from this quantification step. As no overhead costs are occurred within the identified activities, the costs are the same for both methods.

Table 2: Costs for "As is"-business models.

Activity	Costs in euros
Reagent supply	0 (reimbursed)
Test processing	2 / test
Transport	N/A
Laboratory-HIS integration	15/month/practice

As described in the case study, the laboratory sends the reagents to the GP for free. The laboratory does make costs for the reagents, but these are covered by the health insurer and fall outside the scope of the business models. Similarly, a third party handles the laboratory-HIS integration process. The costs for this are the same for both the “As is” and “To be” model. For these reasons, these two processes are not handled anymore hereafter.

4.3.3 Pricing Method

As we are examining the As is-model, the different pricing methods are not applicable. The actual set prices can be used for the current situation. These are directly the costs for the GP, as shown in Table 2.

4.4 To Be Models

The “To be” scenario is an outcome of “step 5: Design alternatives”. The alternative that we examine in this case, introduces MobiHealth’s automated processing of POCT test results, as a replacement of the courier service. In the research for this paper, other more radical alternatives were designed as well (Staring, 2015).

Figure 2 shows the relationships in a simplified e3value model for a situation without courier, but with MobiHealth, “To be.”

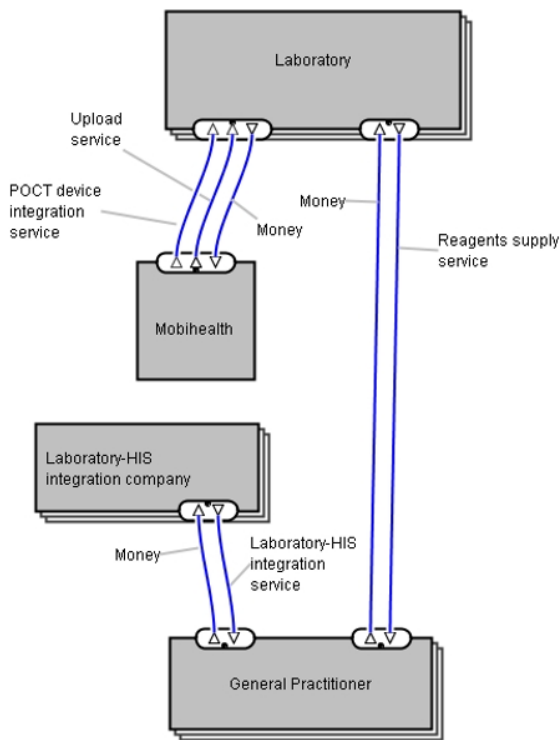


Figure 2: Simplified "To Be" e3value model.

4.4.1 Specify Activities

Similar to the “As is”-situation, four processes are identified for the “To be”-models. The reagents supply process still is the same, yet the other processes differ. The POCT device integration process is initialised by the general practitioner and replaces the test processing service of the laboratory

as the POCT device takes care of testing. The POCT device integration process relates to the POCT device integration service in the e3value model. Tests performed with the POCT devices are uploaded to the laboratory by the test upload service of MobiHealth. Using the existing laboratory-HIS integration, the results are pulled from the LIS to the HIS by the laboratory-HIS integration process.

While different BPMLs were used to model the processes for BMMa (BPMN) and BMMb (EPC), the resulting set of specified activities remain the same (Staring, 2015). Therefore, these models are not included in this paper.

4.4.2 Quantify Model

The costs for the activities specified in the previous step are determined using the methods defined in BMMa and BMMb. Table 3 shows the costs per activity. The costs for the administration of the POCT device integration is for the laboratory, while integration and hardware are costs for MobiHealth.

While different cost accounting methods were used to determine the costs for BMMa (TD-ABC) and BMMb (variable costing), the results were the same. This is logical, since no overhead is visible in the process, and that is where the methods differ.

Table 3: Costs for "To be"-business models.

Activity	Costs in euros
POCT device integration	
Administration	6.25 / device
Integration	50 / device
Hardware	100 / device
Test upload process	N/A

4.4.3 Pricing Method

Both the specified activities and the costs, determined in the previous sections, are used to set the prices using the methods defined in BMMa and BMMb. Table shows the results of this second quantification step.

For the POCT device integration activities, MobiHealth sends the laboratory an invoice for the integration of POCT devices with their systems, the costs for the integration includes time and hardware. From the costs in the previous step, this is estimated to be 150 euros per POCT device.

Table 4: Pricing for "To be"-business models.

	BMMa	BMMb
POCT device integration	100 euros profit margin	0 euros (free)
Test upload process	0.30 euro per upload	1 euro per upload

For BMMa, using the Cost-plus pricing method, interviews revealed a 100 euros profit margin.

For BMMb, using the Value-based pricing method, device integration allows MobiHealth to ensure future revenue by hooking the GPs up to their systems. Therefore, this process is mainly of value to MobiHealth itself and they should not charge for this.

In the test upload process, MobiHealth sends the laboratory an invoice per upload for their services. While the costs for this service are difficult to estimate or allocate, prices may still be set.

For BMMa, using the Cost-plus pricing method, interviews revealed a fixed profit margin. A profit margin of 30 eurocents is used per upload, as costs are estimated to be close to zero.

For BMMb, using the Value-based pricing method, test upload is a high value service for MobiHealth and for the laboratory as this enables the POCT test results to flow to the laboratory. MobiHealth can therefore ask 1 euro per upload.

5 BUSINESS CASE DEVELOPMENT AND ANALYSIS

This section attempts to complete “step 6: Analyse alternatives” of the BMM. We create business cases based on the business models developed in the previous section. To create the business cases, we use the Business Model to Business Case method (BM2BC). It is not an extension of BMM, but rather a method that comes after BMM to further quantify and compare business models (Meertens et al., 2013). It is based on the work of Ward et al., (2008), but is tailored to the development of business cases based on business models. Similar to their work, it has eight components that the business case needs to address.

The created business cases are also the subject of a sensitivity analyses. Since the number of POCT tests performed and the number of POCT devices per general practice (multiple general practitioners can work in the same practice) determine the outcome of the bottom line, different scenarios have been constructed, as Table 5 shows. The scenarios are based on the case study and interviews.

Table 5: Scenarios for sensitivity analysis.

	POCT tests per year	Devices per practice
Scenario 1	720	2
Scenario 2	960	3
Scenario 3	1200	3

Instead of only looking at the bottom line (shown

in Table 6), we used tools for three elements of the business case: effects, risks, and costs. An effects radar shows how the positive and negative effects are distributed among the different business models (Davis and Erixon, 2008). Risks per business model are mapped on a matrix. This clearly shows how risk is shaped overall, and which alternative involves most risk (Bentley, 2010). A costs & benefits radar shows the financial dynamics of each business model. All these tools visually provide more insight in the business cases and the differences per business model.

Table 6: Cashflow per GP over a 5 year period (in euros).

	BMMa		BMMb	
	Lab	MH	Lab	MH
1	5,707.50	1,180.00	3,587.00	3,300.00
2	7,591.25	1,540.00	4,781.25	4,350.00
3	9,479.00	1,900.00	5,975.00	5,300.00

As this paper focuses on the viability of POCT, and the differences between two derivatives of the BMM, only the details applicable for this are handled here. The full details of the research are available in other work (i.e. Staring, 2015).

The viability of POCT in the case study is good. As Table 6 shows, both the laboratory (Lab), and MobiHealth (MH) profit from the introduction of POCT and MobiHealth’s automated processing solution. In the business cases, the difference in pricing method between BMMa and BMMb is translated to the distribution of the benefits for the laboratory and MobiHealth.

6 METHOD EVALUATION AND LIMITATIONS

A pitfall in BMM is that the extensiveness of the first two steps have a large impact on the result. When roles or relationships are missed, this will result in not specifying their activities and they will not be quantified. In some cases, this has to do with the scope at which BMM is applied. In other cases roles and relationships are easily overlooked.

The third step in BMM proved to be very important as it allowed to quantify the business model by showing costs objects and points where revenue was made. The difference between the used methods was not significant in terms of results. Both methods suit their purpose in specifying the activities needed to maintain value exchanges between the roles in the e3value model. However, there is a difference in terms of accessibility of the method. BPMN was more

complete than EPC as the language was richer in elements. EPC on the other hand was easier to understand by non-technical people.

The difference in cost determination in the usage of time-driven ABC and variable costing proved to be non-existent. This is because the specified activities showed no overhead. Overhead is treated differently in the different methods. When the processes are specified in more detail overhead could become visible and the used methods would show a difference. The level of detail at which the processes are modelled might be too high to reveal any overhead in the process. Although any level is better than none and the current models do provide enough insight to perform calculations upon. Still some calculations were unable to be performed as this would require specifics on an individual basis. The calculations for the transport process were neglected as it was highly depended on the distance between the laboratory and general practice as well as the route taken by the courier.

Time-driven ABC proved to be an effective method to calculate the costs, as it was easy to obtain the time it took to complete an activity (simple time measurements). Quantifying the processes specified in the previous step follows a natural path, but is also limits the quantification of the business model. For example, costs on infrastructure, human capital, or other fixed costs may not be covered when following BMM.

The extension with of a pricing method proved to be useful, especially in the development of business cases. The cost-plus pricing method was easy to implement and ensured that costs were covered. The value-based pricing method allowed for a new perspective and focussed on environmental and strategic factors to incorporate into the price. It is important to note that the costs for one role can be the price for another role. However, most costs need to be converted using a pricing method, or are internal costs.

After the business models were developed, the Business Case method for Business Models (BM2BC) was used to create business cases. BM2BC listed the components required to build a business case. However, no relationship or method existed to derive effects and risks from the business models. Therefore, we are unable to verify the completeness of those components. For example, it is unknown whether all effects and risks are covered in the business cases. The most obvious ones have been covered, but the identification of effects and risks is subject to the cognitive capabilities of the researchers. The researcher could (unconsciously) have decided a certain effect or risk is outside the scope the business

model and business case. The cost/benefits component was trivial given the two quantitative steps in BMM. This again proved the value of the added pricing method to BMM. Resource restrictions are not considered a real problem in BM2BC, which is a potential pitfall, as human capital or liquidity requirements are not taken into account. They might come up in the implementation planning stage of the business case, but is should be under consideration during the decision-making in the alternative selection stage.

An assumption that was made, which clearly shows the difference between theory and practice, is that business models and business cases assume only a single business model can be in existence at a time. The “To be”-business models remove the need for a courier, while in practice the courier still plays part in transporting reagents. Not all test can be performed by POCT devices either, requiring the patient to go the laboratory or a courier to transport a test sample (as per the “As is”-business models). In essence, there will always be a combination of “As is” and “To be”-business models.

Concluding, BMM and ultimately the derivatives proved to be a useful method to build business models. It provided a natural logical structured method covering all components of a business model. BM2BC was used to create the business cases because of its focus on business models and provided, similar to any other business case method, a list of components which should be included in a business case. The e3value model proved to be useful. It extended the role-relationship matrix by showing the value exchanges for each relationship. To maintain the relationships, the activities were modelled. In terms of communication to non-technical people EPC was the best choice, but if the goal is to later develop a business support tool BPMN should be used. As the activities showed no overhead, the methods used to calculate the costs showed no differences. In addition, the cost determination is likely to be incomplete as BMM only quantifies the specified activities. The added pricing method made it possible to calculate the benefits in the business case and was therefore essential to be included in the business model. The meta-business model visually summarised the business model making it ideal to communicate to others.

7 SUMMARY

In summary, the findings in this paper are threefold. Firstly, in the case study, all of the business cases for

POCT were assessed to be viable.

Secondly, the basis of the BMM allows for reproducible creation of business models, as two different derivatives came up with the same results, although their calculation and representation is different.

Thirdly, extending the BMM with an extra step, namely selecting a pricing method, showed to be useful for creating business cases.

ACKNOWLEDGEMENTS

The original research for this paper was part of the IntoPOCT project sponsored by ZonMW (The Netherlands Organization for Health Research and Development), as part of the programme “Actieplan eHealth”. Involved partners were MobiHealth (eHealth provider), Reinier MDC (laboratory), and University of Twente. At the time of writing, the POCT service of the case study is provided by Mobipoct B.V.

REFERENCES

- Allee, V. (2008). Value network analysis and value conversion of tangible and intangible assets. *Journal of Intellectual Capital*, 9(1), 5–24.
- Altieri, M. F., & Camarca, M. (2001). Point of care testing. *Clinical Pediatric Emergency Medicine*, 2(4), 275–279.
- Anderson, S. R., & Kaplan, R. S. (2003). Time-driven activity-based costing. *Harvard Business Review*, 82(11).
- Andersson, B., Johannesson, P., & Bergholtz, M. (2009). Purpose driven value model design. In *Proc. CAiSE workshop BUSITAL* (Vol. 9). Citeseer.
- Bentley, C. (2010). *Prince2: a practical handbook*. Routledge.
- Bernstein, J., & Macias, D. (2002). Engineering new-product success: The new-product pricing process at Emerson. *Industrial Marketing Management*, 31(1), 51–64.
- Busse, R. (2001). Expenditure on health care in the EU: making projections for the future based on the past. *The European Journal of Health Economics (HEPAC)*, 2(4), 158–161.
- Davis, L., & Erixon, F. (2008). The health of nations: Conceptualizing approaches to trade in health Care. *ECIPE Policy Briefs*, 4, 1–12.
- Dondelinger, R. M. (2009). Point-of-Care Testing. *Biomedical Instrumentation & Technology*, 43(3), 214–218.
- Drenck, N.-E. (2001). Point of care testing in Critical Care Medicine: the clinician’s view. *Clinica Chimica Acta*, 307(1–2), 3–7.
- Drury, C. (2008). *Management and cost accounting*. Cengage Learning.
- Ehrmeyer, S. S., & Laessig, R. H. (2007). Point-of-care testing, medical error, and patient safety: a 2007 assessment. *Clinical Chemical Laboratory Medicine*, 45(6), 766–773.
- Gordijn, J., & Akkermans, H. (2001). Designing and evaluating e-business models. *IEEE Intelligent Systems*, 16(4), 11–17.
- Gregson, A. (2012). *Pricing strategies for small business*. Self-Counsel Press.
- Hagist, C., & Kotlikoff, L. J. (2006). Health care spending: What the future will look like.
- Kartseva, V., Gordijn, J., & Tan, Y.-H. (2006). Toward a modeling tool for designing control mechanisms for network organizations. *International Journal of Electronic Commerce*, 10(2), 58–84.
- Kiwa Charity. (2014). *Het pad naar duurzame hartfalenzorg: Een praktijkonderzoek naar optimalisatie van het zorgpad chronisch hartfalen met de inzet van telemonitoring* (p. 54). Kiwa Charity.
- Lehmann, C. (2001). Management of point-of-care testing in home health care. *Clinical Leadership & Management Review: the Journal of CLMA*, 16(1), 27–31.
- Meertens, L. O., Iacob, M.-E., & Nieuwenhuis, L. (Bart) J. M. (2012). A Method for Business Model Development. In B. Shishkov (Ed.), *Business Modeling and Software Design* (pp. 113–129). Springer Berlin Heidelberg.
- Meertens, L. O., Starreveld, E., Iacob, M.-E., & Nieuwenhuis, B. (2013). Creating a Business Case from a Business Model. In B. Shishkov (Ed.), *Business Modeling and Software Design* (pp. 46–63). Springer International Publishing.
- Middendorf, I. (2010). *Point-of-care testing impacts outcomes*.
- OMG. (2006). *BPMN 1.0: OMG Final Adopted Specification* (standard No. dtc/06-02-01). Object Management Group.
- Oxford University Press. (n.d.). “business case”. In *Oxford Dictionaries*.
- Pateli, A. G., & Giaglis, G. M. (2004). A research framework for analysing eBusiness models. *European Journal of Information Systems*, 13(4), 302–314.
- Peter, J. P., Donnelly, J. H., & Tarpey, L. X. (1991). *A preface to marketing management*. Irwin Homewood, IL.
- Pouloudi, A. (1998). *Stakeholder analysis for interorganisational information systems in healthcare*. London School of Economics and Political Science, London.
- Scheer, A. W., & Schneider, K. (1992). *ARIS (Architecture of integrated Information Systems)*. Springer.
- Staring, A. J. (2015). Business models & business cases for point-of-care testing. MSc. Thesis, University of Twente.
- St-Louis, P. (2000). Status of point-of-care testing: promise, realities, and possibilities. *Clinical Biochemistry*, 33(6), 427–440.
- Van Dijk, F. (2015). Barriers of Market Entry with POCT:

- A Case Study. In *Proceedings of the TSConIT, University of Twente, Enschede, Netherlands*.
- Vermolen, R. (2010). Reflecting on IS Business Model Research: Current Gaps and Future Directions. In *Proceedings of the 13th Twente Student Conference on IT, University of Twente, Enschede, Netherlands*.
- Ward, J., Daniel, E. & Peppard, J., (2008). *Building better business cases for IT investments*. MIS quarterly executive, 7(1), pp.1–15.

On the Effect of Digital Frontstores on Transforming Business Models

Concept and Use-case from the Consulting Industry

Denis Johann, Tobias Greff and Dirk Werth

*AWS-Institute for digitized products and processes, Uni Campus Nord, Saarbruecken, Germany
{denis.johann, tobias.greff, dirk.werth}@aws-institut.de*

Keywords: Business Models, Business Model Transformation, Digital Consulting, eConsulting Store, Frontstore.

Abstract: Digitization has been changing society and economy enormously for the last 15 years. Surprisingly, consulting, though acting as a key driver in the digitization of other branches, was not subject of any significant changes. The consulting process is still merely supported by ICT technologies. We develop a digital frontstore approach to easily enable digitization of consulting. This approach focuses on the interaction of consultancies with their customers and the support of those using digital technologies, especially ecommerce systems. Such a store requires to transparently and modularly offer consulting services, which is opposed to the current sales model of most consultancies. Hence, we discuss in this paper inherent implications of digitized consulting services on the business model of consulting. In order to assess our findings, we present the actual implementation of an electronic consulting store as initial design artefact. We used this artefact to evaluate our understanding of a digital consulting frontstore. This allows us to conclude our work with a summary of an expert panel reviewing our conceptual findings.

1 INTRODUCTION

Digitization has changed our age. In the last 15 years, it has become the major trend for transforming societal and business life (Kappelman et al., 2014; Von dem Esche and Hennig-Thurau, 2014; Leimeister et al., 2014; Cocorocchia et al., 2016). In this respect, *digitization* means the transformation of products and services that have been originally produced and provided in the real world into computer- and Internet-based services. However, it is not a one-fits-all solution. Each business has to conceive its own approach of their digitization (Veit et al., 2014). This is also - and above all - valid for ICT industry that has been enabling or catalyzing digitization (Sabbagh et al., 2012). Especially those “digital factories” are at risk to be easily surpassed by more digitized competitors. The same applies to the consulting industry that overlaps with ICT industry (Zhou and Muller, 2003). Many consulting companies have profited from digitization for years. They initiated the projects, fostered and propagated them, they were steering and managing them, but they are nevertheless neglecting self-digitization. Smartphones and Excel Sheets are established tools in the consulting domain. But digital software solutions supporting and improving the consult-

ing processes itself are missing putting consulting companies at high risk. They may lose market shares to challenging competitors with novel, digital consulting approaches.

As in the case of ICT enterprises the risk arises primarily by actual success: After the first consulting firms were founded about 100 years ago, consulting industry has been constantly growing (McKenna, 2010). Even though the mergers and acquisitions we have seen in the last years just demonstrates that the market has become mature, there is still an increase in market volume. But maturity implies at the same time that consulting has become a commodity. Looking into digitization development, especially those commodity services are affected, transformed and replaced by new digital business models (Friedrich et al., 2013). To be on top of this development it is highly necessary for challengers and established companies to reveal and realize the potentials and effects of digitized consulting.

Therefore our work contributes to this research field by analyzing and introducing such a potential - the digital frontstore approach - that enables the digitization of consulting service processes.

To introduce the approach we first present our understanding of digitization of consulting - especially

by considering requirements - in the context of already existing solutions for digitized services. Afterwards, we use those insights to develop an initial design artifact of the digital frontstore approach, a lightweight transformation for digital consulting offerings. Analyzing this approach, we can assess inherent implications for the business model of consultancies based on the well-established business model canvas. Our work is concluded by presenting the eConsulting Store prototype as the result of a first design and implementation cycle, followed by discussion with an expert panel with regards to possible implications of our approach to the traditional business model of consultancies.

2 DIGITIZATION OF CONSULTING

Digitization of services has become the standard: In all kind of domains there are efforts to transform conventional services into digital ones (Cocorocchia et al., 2016). This affects retail in the same way as it affects banking and other industries. Thereby, not all services possess the same potential for a digital transformation. With hairdressing – a common example – only the front end part of the process can be digitized: Finding the right hairdresser, fixing an appointment, booking the service and paying it, all this can be operated through the Internet. And there are companies and platforms using such approaches already (e.g. *salonselect.com.au*, *mylocalsalon.com.au*). But the original service provision, the cutting, is and will remain a manual, physical activity that obviously cannot be digitized. In that respect immaterial services are easier targets for digitization since material services will never be completely digitizable. Consequently, services have to satisfy special requirements in order to be digitizable (Greff and Werth, 2015) since they must be – at least partially – transformable into a software solution. Thus, digitizability implies applicability. Consequently, we define digital services objective-based as those services that were originally rendered analogously, but are made more efficient and effective by using ICT. In our context, this holds in particular for digitized consulting

Since personnel costs typically present the largest cost item in consulting business, the standard approach for digitization is to reduce manpower or increase usage of IT (Taherdoost et al., 2013). Ideally, both effects can be combined, although in many cases a human being is only finitely replaceable by ICT. Still in these cases, digitization may help to transfer

processes from the company site to the client site via a proper interface (Werth et al., 2016). Accordingly, staffing requirements and customer contributions are key components to evaluate digital services. This insight enables us to better and more accurately classify digital services, see Fig. 1 (Taherdoost et al., 2013).

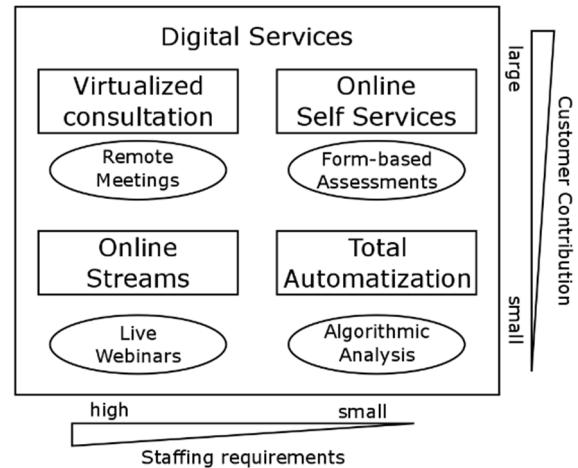


Figure 1: Classification of Services, after (Taherdoost et al., 2013).

This classification also shows, that different technologies are suited to support digital services depending on the quadrant. Consequently, there is no universal technology available for digitized consulting, but a proper implementation is required for every characteristic. They can be coarsely distinguished as given below:

1. *Computer Supported Consulting (CSC)*: Software tools are used to support single tasks of a consultant, while the tools themselves are domain neutral. Only the way they are used provides an added value to the consulting itself. Typical examples are electronic data processing tools.
2. *Computer Assisted Consulting (CAC)*: It incorporates tools developed specifically for consulting economy, supporting specific tasks of this branch.
3. *Computer Controlled Consulting (CCC)*: It extends CAC. Instead of supporting single tasks, a CCC-system assists in rendering the consulting service as a whole. As a key difference, note that such a system focuses on processes rather than supporting specific functions, as it is the case for a CAC-system.
4. *Computer Executed Consulting (CEC)*: CEC aims at completely replacing the consultant. Services originally rendered by the consultant are in this case undertaken by a software platform. The consulting itself is not anymore assisted, but as much

as possible provided by the software. Examples could be automatized audits, as proposed for Self-Service Consulting.

Not all known consulting services can be supported or replaced by any class more sophisticated than CSC, hence current approaches typically focus on electronic data processing tools. It is thus necessary to identify conditions, under which a service can profit from CAC, CCC or CEC towards further digitization. The requirements of standardization, modularization, customization, integration, customer-services and socialization are to be met for a service to be digitized (Greff and Werth, 2015).

A well-known and established example for CAC digitization in consulting is *Quantifye*, which allows to order various consulting services. Here, the potential customer first describes his problems, after which a consultant is assigned to the project based on skills, location and availability. This way, *Quantifye* focuses on mediation of consultants, which prevents the possibility to offer consulting on a small scale directly via the platform itself. Additionally, only the first steps of the consulting process, namely project description and assignment of a consultant are captured by the consultant. *Hillgate* has a similar approach.

A vendor that goes further with a CCC solution is *clarity.fm*, offering a dashboard to chat or talk to a possible consultant. After a consultation, billing is automatically processed on a minutely rate, yet *clarity.fm* does not incorporate customer-site consulting. As our discussions with experts revealed, customers still want the ability to book on-site consulting, as we present in the last section. In addition, the focus of platforms like *clarity.fm* lies on start-ups rather than offering consulting services at any scale.

McKinsey Solutions captures a totally different part of the process and is considered to be one of the most disruptive innovations in consulting so far (Moreau, 2013). Unlike the platforms discussed above, it is a Software as a Service solution that concentrates on providing a broad range of data analysis, processing and presentation tools. As it supports specific consulting tasks it is an additional example for a CAC solution. Yet *McKinsey Solutions* lacks the ability to initiate the consulting process.

Other potential applications of digitized consulting would be remote consulting, for example webinars or video calls. Here, a complete digitization would give rise to more automatization of business

and consulting processes, thus reducing need of personnel and making the consulting service itself more efficient. But the digitization of consulting requires initially changes to the classical web presence of companies, which we discuss in the next section.

3 THE DIGITAL FRONTSTORE APPROACH

Classical online stores like Amazon or iTunes offer either material goods or data including software and content. If such a shop enables a customer to search and buy a product, which is eventually shipped afterwards, the complete purchase process is already captured. A consulting service has a much more complex purchase process, hence requiring a more complex store solution. Therefore we need to clearly define first, how a consulting service is executed from a procedural perspective, see Fig. 2. Broadly speaking, it already starts when a project is specified: A potential customer must align his/her problem with the available portfolio of competences. Afterwards, a consultant must be associated, who has skills matching the project's requirements. In the next two steps, a concept must be developed, evaluated and implemented. The consulting service is completed by evaluation and billing. Each of these steps involves sub-processes that depend on the branch of the actual consulting service, e.g. sales consulting, business consulting or IT consulting. Hence, we do not discuss these sub-processes here in detail. However, it is clear that a solution is needed that can be involved in all major processes above.

If we compare actual websites of consulting companies, they typically lead to a contact form by which a potential customer might only initiate a consulting process. Other than that, several platforms act as mediators between interested clients and freelancer consultants, as we discussed above.

Neither the mediator platforms nor the websites of consultancies are covering the complete consulting process. To get a holistic approach fulfilling this criterion we introduce the digital frontstore (DFS) approach. A DFS is at least a CCC solution, optionally extended by CEC services.

A DFS must allow a customer to search and book services whereas the vendor must be able to combine

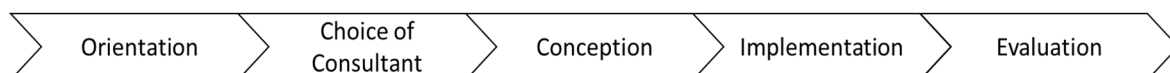


Figure 2: Generic Consulting Process, after (Block and Markowitz, 2000).

and customize services. After booking, services provision must be handled by the frontstore. This handling extends until the service is evaluated after completion. Consequently, we conclude that a DFS must support the following functions:

1. *Electronic Alter-ego*: Both customers and consultants must be represented via electronic alter-egos. This way, the communication between the two parties is digitized and rendered by the platform.
2. *Listing of Product Portfolio*: A potential customer should be able to see all available products offered. In addition, consultants should have the possibility to add services and assign them to their portfolio.
3. *Shopping*: While a customer looks for a service, s/he must first of all filter the portfolio. By the reason of complexity s/he may need support of sales consulting, which must be accessible online, for example via a video call functionality. As soon as a customer has decided to book a service, the DFS offers the possibility to add services to a shopping cart. From this shopping cart, the customer can either check-out immediately or request a quote, especially if s/he books several services.
4. *Payment*: The payment process of the DFS is more complex than for classical online stores. It is typically not the person booking the service, who later has to pay, but rather the financial department of a company. Therefore, the customer must be associated with a company, and billing is finished in contact with the company directly. In addition, for services offered at a minutely rate, payment can only be initiated after service provisioning.
5. *Support of Fulfillment*: Consultants typically render their service in contact with numerous people on customer-side. Henceforth arises the necessity to associate employees of the same company with a booked digital service. In addition, times must be scheduled, eventually with a large team of employees, so the DFS must allow scheduling as well as supporting the fulfillment of a task - especially for those consulting services that are performed in a digitizable way, e.g. calls or webinars. A DFS can even incorporate the whole fulfillment process by offering remote call technology.

The implementation of the listed functions has direct effects on the classical consulting processes. Primarily changes affect sales, scheduling and fulfillment processes (Werth et al., 2016). However within the scope of this paper, we neglect those operational impact of the DFS. Rather we concentrate on the ques-

tion how it affects the business model of a consultancy.

4 EFFECTS ON THE BUSINESS MODEL

Introducing a DFS as described above at first sight only seems to add another channel for sales and customer interaction. Instead of the sales representative, now the customer itself uses the store to select and purchase the consulting services. However, this actually neglects the effects of this new approach on the business model of a consulting company. In this context, a business model is “the business logic of an underlying company by a combination of interdependent offering, market, internal as well as economical business model components in a static and dynamic way beyond the company’s borders” (Burkhart et al., 2011). By selecting this definition we focus on the value creation of a business model. Digitization in particular implies effects on the value creation (Methlie, 2000; Weill and Woerner, 2013; Johnson et al., 2008; Moreau, 2013). The identification of the effects in case of the DFS shows the real added values, justifying the DFS approach in particular and digitization in the consulting domain in general.

Effect 1: The Long Tail A common and often referenced effect of an e-shop solution is the long tail (Oestreicher-Singer and Sundararajan, 2012; Elberse 2008). Amazon as a well-known example highly profited from this effect (Elberse, 2008). It mainly means the improved possibilities to offer products and services that are sold rarely. The positive effect on the value creation results from the mass of such products, that individually are not producing high sales volume, but the pure mass of them provokes significant revenues. In our context, this means that consultancies may offer a large variety of modular services, reflecting their actual skill portfolio, as an extended offering. It enlarges the limited offerings which sales consultants usually present in on site customer appointments. This is in line with the second effect:

Effect 2: Explicit Offering Consultancies usually offer a wide range of problem solving competencies. Similar individual consultants rather present themselves as capable of solving any problem in their area of expertise. The scoping and detailing of the consulting activities is discussed and negotiated within bilateral meetings with customers. However, this procedure is rather less compatible with the functioning of an online store. The catalogue structure of such a

store requires the definition of the products (and services) to be included in the catalogues. In this respect, it becomes mandatory for consultancies to switch from a “we solve everything” into an explicit offering approach. They need to specify precisely what consulting services they offer, how they are fulfilled and how much effort this requires.

Effect 3: Price Transparency This effect also results from the catalogue nature of an online store.

All services are offered at a fixed price and can be easily compared by potential customers in contrast to the current model of consultancies to only provide request forms and not publically revealing prices. Thus, this comparability already changed the pricing systems of several industrial branches and gave rise to comparison platforms (Weinhardt et al., 2009; Klein and Bhagat, 2010) and may even more affect consultancies since their pricing model heavily relies on competitors’ prices (Ivero et al., 2013).

Effect 4: Pre- and Post-consultancy Automation

Online stores change the sales processes and concepts of consultancies. They benefit from the fact that the sales cycle is reduced tremendously, since the time to set up an offer is negligible. The customer itself browses through the service portfolio and selects appropriate consultancies. But the automation (from the point of view of the consulting company) is not limited to the sales process. Also other auxiliary tasks can be executed electronically without human intervention. Examples are the determination of appointments, or the invoicing. In this situation, the online store can incorporate the complete pre- and post-consulting process, not only saving time but also increase comfort for customers and consultants via its increased flexibility, a one-fits-all interface and extended reachability. Consultants and specific consults can be searched online at any time, independent of specific business hours.

Effect 5: Micro Consultancy The introduction of an electronic channel for purchasing consultancy services can change the way how pre- and post-consulting is processed. But can it also change the consulting services itself? It has already been revealed that remote consulting can be offered at significantly lower prices than on-site consulting (Oracle) High wages in consulting are typically a result of the large amount of time spent on and the employees’ willingness for travelling. But the consequences of removing travel costs and automating pre-and post-processing efforts are going farther: The overhang of a consulting service can be reduced enormously. When a consultant does not need to travel to the customer and if the sales and invoicing costs are marginal, there is no need to just sell long time periods to the same customer. Even

providing hour-wise consultancies now becomes economically possible. Moreover, the role model of consultants changes from spatial reachability towards temporal reachability.

Effect 6: Standardized Services In addition to the effects already mentioned, it would be possible to just push the existing consulting offerings to the eshop. However, this would not result in the desired objectives. E.g. there is an inherent risk if offering fixed price services without being clear in the operations. Therefore, in order to catalogue the offered services, there is a minimal requirement of standardization regarding the types of services offered, the description of the actual service and how fulfillment is done (Gottschalk et al., 2002).

Effect 7: Modular Consulting Incorporating those effects leads to new options in the design of consulting services. Having a larger portfolio of specific and standardized consulting services, companies can use those to combine them and bundle them into new offerings. In the same way, existing services can be reviewed and transformed in a way that they (at least partially) are composed by other, smaller components. Finally, this results in a modular design of consulting services. And this effect is not limited to their own services: Consulting companies can easily augment their service portfolios by offering or bundling additional products or services provided by third parties along with the consulting services, for example guides or software packages.

In order to provide a more structured view on the influence of the DFS on consultancies’ business model, we use the business model canvas (Osterwalder and Pigneur, 2013). This scheme decomposes business models in individual areas of concerns. Even if those areas are not fully independent (Krumeich et al., 2013) it enables a depicted analysis of the factors which are affected by the DFS. The resulting canvas is shown in Fig. 3. It contains the effects mentioned above and the consequences derived from the conditions of a DFS. For example software and hardware as resourcesE2. It is remarkable that all sectors of the canvas undergo an adaptation due to the DFS. This reflects the significant impact of the approach on the business model of a consulting company.

5 THE ECONSULTING STORE USE CASE

The concept of the digital frontstore approach developed above was implemented as a prototype called










<p>Key Partner </p> <p>0 not visible on platform 0 sub partner network in the backend</p>	<p>Key Activities </p> <p>+ integration to a holistic approach + platform built + hybridization with classical business</p>	<p>Value Proposition </p> <p>+ explicit offering (E2 & 6) + extended portfolio (E1) + price transparency (E3) + cost efficiency (E4 & 5) + micro consulting (E5) → time independency + reachability (E2) + modularity (E7)</p>	<p>Customer Relationships </p> <p>+ remote assistance + self services + automated services + virtual organization 0 co-creation</p>	<p>Customer Segments </p> <p>Diversified customers: + customers willing to use web-based services + efficiency oriented mass market (E1) 0 existing customers with framework contracts</p>
<p>Key Resources </p> <p>+ software + hardware + standardized service (E6) catalogue / modularity - human resources</p>		<p>Channels </p> <p>+ webchannel: purchase, scheduling, fulfillment, sales consulting (E4)</p>		
<p>Cost structure </p> <p>+ IT costs / SEO optimization - travel expenses - personnel cost</p>			<p>Revenue Streams </p> <p>+ fixed-price consulting services + small scaled consulting + upselling + reduced sales cycle</p>	

Figure 3: Effects of the frontstore approach on the consulting business model classified by business model canvas. (“+”/”-“: increase/decrease of the respective item, “0”: item unaffected).

eConsulting Store (ECS) together with the consulting company Scheer GmbH. Implementing all requirements of the DFS would require a large complexity, making it more expedient in the short term to evolutionary digitize single stages of the consulting process. The resulting modular solutions may be merged later into a complete DFS. This must be kept in mind during the initial design cycle. Hence, we focus here on designing a quickly applicable solution providing an entry point to digital consulting.

The prototype was implemented and customized for the company specific IT Consulting Services. The respective business unit provides consulting services for a specific software family. The offered consulting services are characterized by their close relation to this specific software and its structured rollout, customization and usage. We could henceforth assume, that a digitization of the consulting service is principally possible, since no spatial or functional binding of that consulting services exists. As the offered consulting services aim at IT-based solutions, we could also assume an optimal willingness of potential customers to work with new technologies, which is the reason why we chose this business field for a prototypic implementation.

After verifying suitability we identified needs in this field in form of a use case. We found that custom-

ers in particular demand internet-based booking, processing and billing of two different consulting services. The first one being workshops, classified as one-to-many services. The second one are modular one-to-one consulting services. In both situations, the client must be able to buy and process the service remotely. As an example for such a one-to-many consulting workshop we consider the “BPA workshop”-offering of the consulting company. Here, all of the customer’s current BPA- and BPM-activities are determined within four days and possible opportunities of optimization are evaluated. As an instance for one-to-one consulting services we reference to complex live business model development and modelling assistance with business process tools.

As a proof of concept, we implemented the ECS in the domain of those IT Consulting Services based on the open source webstore Magento. Our choice was governed by accessibility of the source code, variability of available plugins, customizability and general extensibility by individually developed modules. A first result was a customer oriented architecture of the eConsulting Store that mainly consists of two components (see Fig. 4). The frontend essentially handles purchases while the backend bundles the organizational process surrounding consulting services. A user management needs to span both components, since user registration and verification is required,

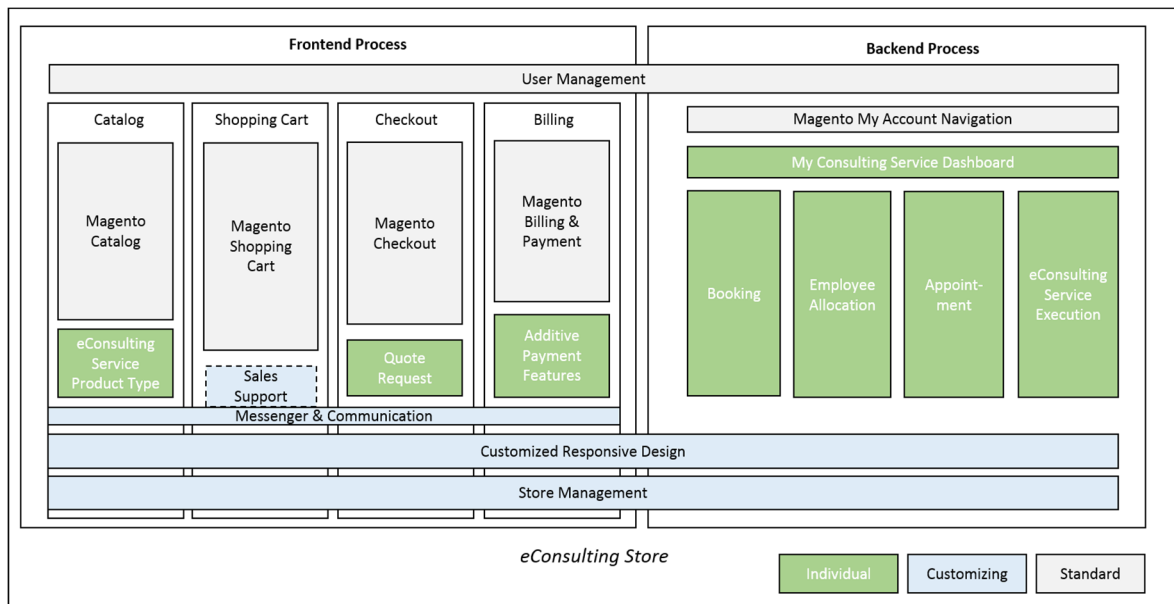


Figure 4: eConsultingStore architecture.

Responsive Design is implemented to assure presentation of the store independent of the user’s device. In addition to client-oriented components, we have an administrative Store Management, as an interface to the provider to update offers and adapt the store in detail.

We provide the details of the prototype by the typical use-cases mentioned above, namely a customer searching, ordering and scheduling a consulting service - be it a workshop or a one-to-one consulting service - as well as fulfillment and after-sales services.

At first, a potential customer sees the home page of the web store, explaining the concept of the store. From here, s/he can navigate via the menu bar to a browse-functionality, where the customer can either browse through all available services or search for a specific service by keywords. From the overview, the client can request detailed information about specific services by simply clicking on the product of interest. From the detail-view, the customer can now add a service, including the volume of services needed, to his shopping cart. If customers are unsure about which products suit their individual needs, they may request an online sales consulting via a live chat function. Besides standard functionality of a shopping cart like removing items and checking out, we set up a function to request a discount, if for example several instances of the same product are purchased.

If a customer decides to buy a service directly, the buying process is invoked, during which s/he may choose the preferred method of payment. Besides classical payment methods like credit card payment,

we allow the customer to be associated with the financial department of a company. Often it is not the actual customer booking the service, who later pays the bill, but rather the respective employer.

If now the cart is checked out, a database object is generated on server-side. All earlier bookings can be accessed by shop administrators through a dashboard. Nonetheless, the billing process is started automatically after the purchase is finished.

A customer may access the booked and payed services via a GUI. Here, s/he can schedule services, for example in the case of a call. Additionally, s/he can associate other users of the same company with a service, for example by inviting them to a webinar.

If we now look at the ECS, we find that it matches most criteria of a digital frontstore that is able to perform digital consulting services. Orientation, selection, conception, and evaluation are fully implemented. Only the implementation is not fully available, since the service itself is in most cases still performed outside the scope of the platform. Although hybridization is part of the platform the choice for on-site or remote consulting should lie on the side of customer. Therefore, this is going to be part of our future work.

6 EXPERT REVIEW

During the initial stage of the ECS, we collected opinions on changes and effects of a DFS on consulting processes during roadshows, talks, conferences and

conceptual stakeholder meetings. As a majority, these opinions reflect biased positions of the involved roles, containing statements by experts as well as directly affected people. The opinions are sorted by five aspects, representing compact catchwords for a specific change caused by digital consulting. We found that the experts in general agreed with our expectations as discussed in section 4. In addition, our experts brought forward the following aspects:

1. *Spatial Independency*: “The customer must always be able to choose between remote or on-side consulting.” “A success of remote-consulting depends on comprehensible win-win situations.” Examples here are significant discounts based on savings in travel costs. “A sustainable model could incorporate hybrid types in the context of a consulting project. A first get-together on client-side followed by remote-consulting.”
2. *Small Scaling*: This means, that the ECS can offer smaller consulting units in the context of duration and extent. “This allows to even reach small markets.” But not all of these markets are profitable. “Addressing a broader customer base, for example by internationalization, is therefore considered a necessity.” According to consultants’ opinions, a smallest subunit is still inevitable. Two or three topics a day are considered to be the maximum. “Changing topics too quickly is considered to endanger the consultant’s focus.” Disadvantages are expected by losing a negotiating basis and enable comparability with competitors.
3. *Reachability*: This is a surplus value especially for customers. New digital and existing bookable consulting services are permanently viewable and reachable. Consulting becomes a multi-channel offer. “If every channel here comprises an added value is hard to estimate.” “Synergies for sure add value, but also the risk to neglect essential channels.” “Digital consulting does not assure the contact to a specific consultant, but it needs to be guaranteed that a consultant with equivalent skills is available remotely or locally.” The fact the recipient of a consulting service is different from the buyer is considered to inhibit the reachability and usage of digital channels.
4. *Analyzability*: “Digital consulting promises a large data analysis potential by CRM, structured collection of feedbacks and even predictive analytics for customer needs.” In particular, this allows automatized refinement and presentation of success stories. In it highly controversial but often assumed, that “analyzability by digital consulting comprises security for the company, since con-

tacts and projects are less cross-linked and undergo smaller fluctuations.” The largest concern in the context of analyzability is “inhibition due to doubts about data security.”

5. *Reusability*: “Tools from digital consulting support consulting processes today – and will do so in the future – in different hybrid forms wherever and whenever redundancies are evitable.” Webinars are an example of a product that can be consumed several times or the construction of a knowledge management system, assuring long-term preservation and usage of knowledge. “Especially training of new employees or exchange of knowledge within a company offer large potentials.” A problem of reusability is of course the need for a copy protection.
6. *Scalability*: With this we think of reaching arbitrarily many customers with appropriate effort. A concise example is that of a “remote consulting by one consultant, rendering his service to hundreds, if not thousands, of customers.” “This would be a fair model for trade show consulting, but personal consulting is hard to imagine due to a large variety of individual aspects.” Rather than that, the idea is to make the business model of digital consulting scalable. Digital consulting should offer the optimal mediation of consultants. “Mediation as a service can be arbitrarily scaled.” “Computer-assisted networks, incorporation of freelancers and generation of a broad employee base by breaking-up traditional business structures are essential driving forces of this idea.” A concern here is mediating credibility. In addition, digital mass consulting comes with a comparability that is considered to be critical.

7 CONCLUSION

In this work, we introduced a digital frontstore approach for consulting services. Using a business model canvas, we identified impacts on the business model of consultancies and found that a DFS mainly affects all components of a consultancy’s business model. We provide as a first design principle the eConsulting store, allowing us to discuss the DFS approach with an expert panel.

Since the ECS is the result of a first design cycle, future work is going to include implementation of further functions of the DFS, as discussed in section 3. In addition, future theories and prototypes should take into account the results of our expert panel.

It will be interesting to match our expectations on

how a DFS may affect the business model of consultancies with its actual impact on the business model of consultancies, as soon as a full DFS solution is available.

REFERENCES

- Block, P. & Markowitz, A., 2000. *The Flawless Consulting Fieldbook And Companion: A Guide Understanding Your Expertise Author: Peter Block, Andrea Markowi*, John Wiley & Sons. Available at: <http://www.openisbn.org/download/0787948047.pdf>.
- Burkhart, T. et al., 2011. Analyzing the Business Model Concept— A Comprehensive Classification of Literature. In *Thirty Second International Conference on Information Systems*. pp. 1–19.
- Cocorocchia, C. et al., 2016. *Digital Media and Society Implications in a Hyperconnected Era*, Geneva.
- Elberse, A., 2008. Should You Invest in the Long Tail? *Harvard Business Review*, pp.1–9.
- Von dem Esche, J. & Hennig-Thurau, T., 2014. *German Digitalization Consumer Report 2014*, Münster.
- Friedrich, R. et al., 2013. *The 2012 Industry Digitization Index*, London.
- Gottschalk, K. et al., 2002. Introduction to Web services architecture. *IBM System Journal*, 41(2), pp.170–177.
- Greff, T. & Werth, D., 2015. Auf dem Weg zur digitalen Unternehmensberatung. *IM + io - Das Magazin für Innovation, Organisation und Management*, 1, p.30.
- Iveroth, E. et al., 2013. How to differentiate by price: Proposal for a five-dimensional model. *European Management Journal*, 31(2), pp.109–123.
- Johnson, M.W., Christensen, C.M. & Kagermann, H., 2008. Reinventing Your Business Model. *Strategy*, 86(12), pp.57–68.
- Kappelmann, L. et al., 2014. The 2014 SIM IT Key Issues and Trends Study. *MIS Quarterly Executive*, 13(4), p.237.
- Klein, A. & Bhagat, P., 2010. We-commerce: Evidence on a New Virtual Commerce Platform. *Global Journal of Business Research*, 4(4), pp.107–124.
- Krumeich, J., Werth, D. & Loos, P., 2013. Interdependencies between Business Model Components—A Literature Analysis. In *AMCIS 2013 Proceedings*. AIS.
- Leimeister, J.M., Österle, H. & Alter, S., 2014. Digital services for consumers. *Electronic Markets*, 24(4), pp.255–258.
- McKenna, C.D., 2010. *The World's Newest Profession - Management Consulting in the Twentieth Century*, Cambridge: Cambridge University Press.
- Methlie, L.B., 2000. A Business Model for Electronic Commerce. *Teletronikk*, 96(2), pp.8–19.
- Moreau, F., 2013. The disruptive nature of digitization: The case of the recorded music industry. *International Journal of Arts Management*, 15(2), pp.18–31.
- Oestreicher-Singer, G. & Sundararajan, A., 2012. Recommendation Networks and the Long Tail of Electronic Commerce. *Mis Quarterly*, 36(1), pp.65–83.
- Oracle, Oracle Homepage. Available at: http://www.dba-oracle.com/consulting_prices.htm#oracle [Accessed March 15, 2016].
- Osterwalder, A. & Pigneur, Y., 2013. *Business model generation: a handbook for visionaries, game changers, and challengers*, John Wiley & Sons.
- Sabbagh, K. et al., 2012. Maximizing the Impact of Digitization. In *The Global Information Technology Report 2012*. pp. 121–133.
- Taherdoost, H., Sahibuddin, S. & Jalaliyoon, N., 2013. E-Services Usage Evaluation ; Applications ' level of Co-Creation and Digitalization. *International Journal of Academic Research in Management*, 2(1), pp.10–18.
- Veit, D. et al., 2014. Business Models. *Business & Information Systems Engineering*, 6(1), pp.45–53.
- Weill, P. & Woerner, S.L., 2013. Optimizing Your Digital Business Model. *MIT Sloan Management Review*, 54(3).
- Weinhardt, C. et al., 2009. Business Models in the Service World. *IT Professional Magazine*, 11(2), pp.28–33.
- Werth, D., Greff, T. & Scheer, A.-W., 2016. Consulting 4.0 - Die Digitalisierung der Unternehmensberatung. *HMD Praxis der Wirtschaftsinformatik*, 53(1), pp.55–70.
- Zhou, Y. & Muller, L.M., 2003. Technical, Managerial, and Organizational Changes in Adapting an E-Commerce Model in Civil Engineering Design and Consulting Services. *Leadership Manage. Eng.*, 3(4), pp.202–212.

3D Printing and Additive Manufacturing Capability Modelling

Vaughan Michell

University of Reading, Whiteknights, Reading, U.K.

v.a.michell@henley.ac.uk

Keywords: 3D Printing, Capability, Affordance, Capability-Affordance Model, Additive Manufacturing.

Abstract: The use of 3D, or additive manufacturing, is becoming more widespread and is seen as a new industrial revolution due to the advantages of a material deposition approach compared to material removal. However, little work has been done to identify and formalise the capabilities of this new technology. This paper formally analyses the generic 3D printing process of additive manufacturing and compares it with the traditional subtractive manufacturing process using the capability affordance model to determine its unique capabilities. The CAM model defines a capability as a mechanism and space-time path. Results show that whilst the mechanisms differ in terms of force and heat drivers, it is the space time path topology that is key to manufacturing capability differences. We apply a topological analysis to identify the unique affordance path of 3D printing which clearly demonstrates its superiority in complex and integrated part manufacture. Finally we outline the differences in the key capability affordance factors for manufacturing in the two methods. This paper builds on earlier work concerning the capability affordance model as a knowledge model to analyse and understand capabilities and the unique advantages and possibilities of 3D printing.

1 INTRODUCTION

Rapid prototyping (RP) is an additive manufacturing technology based on the addition of materials layer by layer instead of traditional cutting and removal of material in subtractive manufacturing (SM) (Berman, 2012). RP was developed to provide a general fabrication machine enabling construction of complex 3 dimensional (3D) shapes and to use designs directly from CAD (Choi and Samavedam, 2002). The term rapid prototyping – RP is often used to describe the rapid fabrication of parts and prototypes layer by layer (Berman, 2012). The term additive manufacturing (AM) is used to generally denote technology where fabrication of parts and products occur by adding or depositing material.

Three dimensional printing (3DP) is really a versatile consumer adaptation of rapid prototyping technologies evolved from additive manufacturing by adding a 3rd vertical z dimension to the general architecture of a typical x-y inkjet or laser printer. This greatly reduces the cost of the RP/AM machine by several orders of magnitude, making it affordable to companies and end users alike. Instead of ink, 3D printing deposits material, usually hot plastics or their derivatives. The print head thus becomes a means to deposit the material as a fluid deposition of 3d

cylinders or ‘slugs’ of material in successive layers (Berman, 2012). Pham and Gault (Pham and Gault, 2012) identified up to eight different technology approaches to additive manufacturing. For comparison purposes this paper focuses on a frequently used 3DP technology where a heater melts a suitable material and temperature sensors ensure the correct flow viscosity. To enable a continuous feed of material a series of rollers is used to drive the plastic filament at the correct rate to be melted.

1.1 3D Printing Fabrication

3D printers enable CAD models to be converted to a series of layers that can be printed one layer at a time. This layer based model is typically in the form of a stereo lithography or STL file. The STL file records the surface shape/section of each layer, The internal structure is reduced to a series of diagonal webs to reduce both the volume and density of the construction and the time taken to print it compared to a solid layer. STL files can be generated from traditional CAD packages by electronically slicing the design.

The key to 3DP business success is reducing costs and complexity of part creation whilst integrating with consumer software (Pham and Gault, 2012). For this reason printing plastics have focused on

thermoplastics PLA and ABS (Bassett et al., 2015).

However thermoplastics have their own limitations of strength and robustness as PLA suffers from degradation. The 3D object is built by depositing material from the minimum of a point ‘blob’ to an x-y plane of material. Moving the bed on which the material is deposited along the perpendicular z axis away from the print head enables the model to grow vertically. Additional control of the deposition process is usually achieved by heating the build bed or enclosing the build volume with a case to optimise the conditions for solidification without cracking.

Another fabrication benefit is the ability of a 3D laser scanner to digitise the spatial position of points on the surface of a scanned object. The CAD compatibility enables a 3D model to be built from data capture of the surface of any scanned object. This enables scanned object models to then be quickly scaled and printed out on a 3D printer subject to the size limitations of the deposition mechanism used and the build bed in a form or ‘scanning to production’ of a finished part.

3DP has proliferated among manufacturing companies, designers, end users and hobbyists as it meets attributes of successful innovation as it is cheaper than subtractive manufacturing, compatible with CAD/CAM, less complex, easy to try out and produces rapid end results. (Pham and Guilt, 2012).

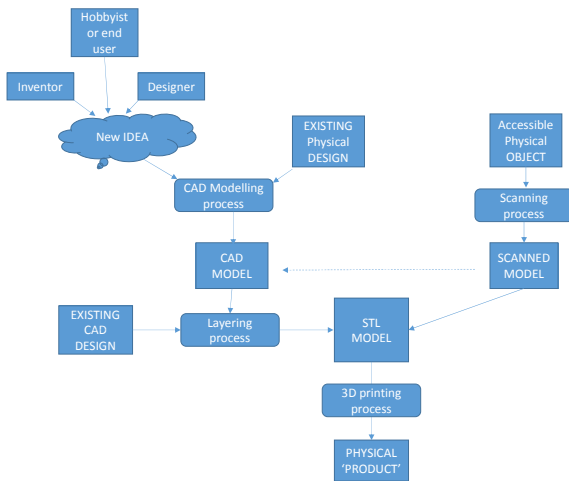


Figure 1: Creation flow using 3D printing.

Any new product idea can be quickly rendered in CAD and then fabricated after STL conversion on a 3D printer or scanned and then printed. This greatly reduces the complexity of the design and creation process as seen in Figure 1.

1.2 Paper Motivation and Layout

3DP is widely reported as a game changer that will alter the world in terms of product supply, manufacturing and retail (D'aveni, 2013). Whilst there are limitations due to low speed of fabrication and robustness of the materials used these are rapidly being overcome (Nair, 2014). This is aided by the rapid and wide use of AM/3DP among industry, designers and hobbyists and consequent reduction in unit costs driven by innovation. It prompts two research questions; a) What are the perceived business and technical advantages of 3D printing? b) How can we compare and model the capability of 3D printing with traditional subtractive manufacturing? To answer these questions our paper sections cover:

- The background to 3D printing and its structure
- Advantages of 3DP from a business and technical perspectives
- Subtractive manufacturing and limitations
- Capability and affordance modelling
- Mechanism Modelling
- Path Topology Modelling methods
- Discussion of findings/conclusion.

2 3DP PROCESS CHARACTERISTICS

2.1 Business Advantages of 3DP

The benefits of 3D printing lead to major business advantages, where a business advantage is defined as superiority in delivering a strategic business benefit.

3DP/AM offers the benefit of a single general production tool compared to a myriad of machine tools required in subtractive manufacturing. 3DP/AM develops this advantage due to its 3 axis and point deposition versatility enabling the creation of complex parts all on one machine. This removes the need for sequential machining processes and jiggling and tooling (Choi and Samavedam, 2002) necessary to hold parts firm against the cutting force implicit in subtractive manufacturing.

3DP/AM also provides the opportunity for integrated design to manufacture via as the software format of 3DP has evolved from, and was deliberated designed to integrate with CAD models (Choi and Samavedam, 2002).

3DP has a unique benefit of rapid mass customisation from creator to end user. The designer and/or the end user is able to rapidly design parts in well-established CAD formats. High quality and fast

fabrication can also be cheaply outsourced by sending it to company to print anywhere in the world.

This greatly simplifies and reduces the cost of the traditional mass customisation models that require complex processes and equipment, for example used by companies like Dell (Berman, 2012). This enables any customer to create a new product, design or adjust their product off line and send it directly to a manufacturer in a ‘democratisation of the design process’ (Hoy, 2013).

With 3DP/ the number of devices required to enable design to production is reduced to one low cost unit, reducing the learning curve and start-up costs, hence encouraging adoption. (Nair, 2014).

2.2 Technical Advantages of 3DP

We define a technical advantage as an advantage in the process, or mechanism of use, of a technology. One of the critical advantages of AM over traditional material removal and moulding methods is the ability to develop unique shapes. This property is a function of the topology of the printing device due to its point deposition of material so that objects can be built in single 3D points and layers at a time (Rosen, 2007).

The creation of a part from a single point enables almost any shape to be built (Rosen, 2007) subject to the need for support at overhangs and specific angles and fine topographies on certain 3DP process types. This differs from SM processes that require extensive force application to remove material via traditional lathe and milling machinery. AM deposition is relatively force free apart from the weight of deposited material that can distort the design when setting. Depending on the resolution of the print head and materials used, this enables extremely fine, small and optimally engineered parts to be produced without the force problems caused by material removal. The point disposition ability at any location within a build volume, coupled with digital design, also enables parts to be made ‘within a part’ as there is no necessity for cutter insertion and removal space as in traditional manufacturing. For example the creation of ball type bearings inside a complete ball race can be executed in a single AM operation (Conner et al., 2014). The point deposition ability of AM enables sub-assemblies and products to be built as complete volumes and surfaces, significantly reducing the need for assembly and reducing costs. For example a complete nylon ball bearing assembly can be produced in one operation as one part vs 18 separate parts assembled using traditional manufacturing due to SM machining limitations (Conner et al., 2014).

The 3DP point deposition ability further enables parts to be created as a fine mesh or space frame, greatly reducing weight, in contrast to machining from solid. This enables maximisation of strength per unit volume (Pham and Guilt, 2012). The ability to deposit in points and layers also supports integrated materials structure variations within a part and the creation of cellular materials to support energy, thermal and acoustic design variations (Rosen, 2007). This enables ‘designed property gradients’ where the density of the product can be designed to vary to suit design needs (Rosen, 2007). 3DP further enables the creation of generic ‘graded’ ‘cellular materials’ such as lattices, honeycombs etc to suit design needs for density and other physical property variations (Rosen, 2007).

A potential revolutionary advantage of the material deposition technique is that it enables the embedding of different material types or even whole components as the part is fabricated. This facilitates variation in composition of the object material properties and additional elements such as areas of electrical conductivity or embedded components to suit design needs (Doubrovski et al., 2011).

3 SUBTRACTIVE MANUFACTURING (SM)

3.1 Limitations of SM

This section explores the traditional subtractive manufacturing process of material removal using a range of different machine tools based on different cutter topologies.

The capability of traditional machine tools are limited by the cutter path which is determined by the cutter shape and mechanics. For example a lathe has two degrees of freedom about an axial symmetry, resulting in a capability for producing axially symmetric parts. A milling cutter has three degrees of freedom about x, y, z axes to produce a diverse range of convex and concave parts. In contrast a drill has effectively only one degree of vertical freedom to create a hole of varying depth.

In traditional SM the lack of a universal cutting machine and limitations of cutter path and finite cutter size means the order in which a complex part is manufactured must be considered to avoid unwanted tool interactions (Matthews, 2007). This order of processing (OOP) limitation adds complexity by imposing machining precedence constraints and conditions on the direction of tool application or

approachability, limiting what is feasible to fabricate (Gupta et al., 1997).

One complication is accessibility (AC), where the arrangement of certain shapes drives the order of machining, otherwise there would be no cutter access.

Gupta et al., (Gupta et al., 1997) defined feature accessibility as a condition where the volume of space required for accessibility A_v has not been lost by a volume already removed from the workpiece.

For example, if the accessible volume of a piece of material X intersects with the removal volume of Y , the cutter must approach X through the volume occupied by Y and hence Y must be machined before X (Matthews, 2007).

Also if X and Y have different approach directions, and machining X provides a surface to act as datum for measurement or tolerance, then X must be machined first. This is a Datum Dependency (DD) limitation of SM (Matthews, 2007).

Another concern is approachability (AP): if X is machined before Y then Y will not be able to be approached or accessed to create the shape (Gupta et al., 1997). For example eg a hole may need to be machined first to provide access to cut a slot.

3.2 Cutter Path Limitations (Cpl)

At the fine scale level for SM we must also consider limitations due to cutter size and geometry.

Whilst convex parts can be relatively easily machined with SM, concave parts eg edges and pockets or blind holes to be cut into the part are limited by the need to ensure cutter access.

Gupta et al (Gupta et al, 1997) defined three types of limitations for cutter path access:

- CPL1: The curvature of the tool must be less than the curvature of a concave edge – otherwise the tool cannot follow and cut the edge (d_{cur});
- CPL2: Where two closed edges form a narrow passage to be cut by the tool the diameter of the passage must be less than the tool (d_{pas}) to enable the cutting of the passage by the tool;
- CPL3: Concave corners formed by the cutter path must be larger than the tool (d_{cor}) to enable cutter access;
- CPL4: as the tool size increases the cutting trajectory (ie the minimum amount of arc that you can cut with this size cutter decreases ie is a limit on d_{tra}) beyond which you will cut a bigger arc than specified;
- CPL5: The max diameter of tool also must be less than the smallest diameter feature (min d_{max}).

This results in Gupta's set of cutter path trajectory

limitations for the maximum cutter diameter:

$$d_u = \min (d_{cur}, d_{pas}, d_{cor}, d_{tra}, d_{max}) \quad (1)$$

Some topologies eg producing a part within a part (PP) using subtractive manufacturing are almost impossible with SM as it requires a tool to enter the interior of a billet of material and then the operation of that tool via a cutter path wholly within the billet.

The geometry of the cutting tool with respect to the part is also subject to two further cutter geometry (CG) limitations. The tool cutting edge angle with respect to the cutter motion α and the inclination of the tool cutting edge with respect to the part, ie the rake angle β must be within ranges to avoid skimming or gouging the material and adversely affecting surface quality (Blackenfelt, 2001).

3.3 SM Conceptual Model

Based on the above discussion we can now develop an outline conceptual model for subtractive manufacturing. Consider the following: where P_s is a part to be created by the subtractive manufacturing process, B = material billet from which the part is machined, and F_i is the i 'th machined feature (eg hole, passage, curved surface etc) where $i = 1, \dots, n$. Then the part is the billet minus the union of all the features.

$$P_s = B - \bigcup F_i \quad (2)$$

But each of the i in number features are topological elements that in SM are created by the intersection of the swept volume of the j in number cutter operations S_c required to machine the features with the billet volume B such that:

$$P_s = B - (B \cap \sum_{(j=1-n)} S_{c_j}) \quad (3)$$

But as we have seen the swept volume is limited by cutter approach geometry, size and access limitations and the prohibited volumes discussed earlier. Hence a generic model for traditional machine tool material removal manufacturing is:

$$P_s = B - (B \cap \sum_{(j=1-n)} S_{c_j}) - CPV \quad (4)$$

Subject to the constraints that the j cutter paths are processed in the appropriate order and within processing constraints (AC, DD, AP, CG) and cutter path limitations (CPL). Where S_c is the swept volume of cutter, CPV are the cutter prohibited volumes, such as a part within a part etc. We will reuse this equation later, but we now focus on how we can identify and measure capability.

4 CAPABILITY & AFFORDANCE

4.1 Overview

Capability is a function of an action process, and the nature of the interaction between two or more objects or resources (Michell, 2011). Capabilities can be modelled by understanding the way possible actions or affordances can or cannot occur. Gibson (Gibson, 1979) defined affordance as the ‘property that the environment or physical system offered the animal to enable a possible useful transformation for the benefit of the animal’. Affordances refer to descriptions of (verb-noun) object abilities such as a road is ‘walkable’ or the ‘cup affords drinking’ (Gibson, 1979) indicating that the structure/disposition of a road or cup– enables it to be walked on or drunk from. Affordances focus on the possibilities of how the object could be used by the animal or person. Affordances require a driving agent, an animal or natural process for them to occur. Turvey’s affordance model related animal properties Z and properties of other entities X in an environment and showed that affordance depended on the state or properties of the animal/object and their ‘dispositions’ (Turvey, 1992). Using the concept of affordance we developed a model of capability – the capability affordance model (CAM) (Michell, 2012).

The model decomposed the affordance-effectivity disposition into (i) a causal energy mechanism eg force, temperature or electrical difference that enables, the capability to occur. This energy flow is transferred or dissipated through a (ii) space-time affordance path that varies with the characteristics of the animal and objects used in the interaction (Michell and Roubtsova, 2014).

As affordances require animal or natural changes to occur to make them realisable we define the affordance mechanism as the cause and effect transformation at the interface between two or more interacting resources and its properties that enable the transformation (Michell 2013). Mechanism thus refers to the behaviour and properties of the energy transfer that drives the transformation. For example a potential difference enabling an electric motor to rotate, or human energy is transferred to enable the capability of a manual device like a syringe to ‘afford’ injection. Affordance mechanisms AM can be typically modelled as force, heat or other energy equations. However, for comparison purposes it is not necessary to include the detailed mathematics, only to understand their differences. The affordance path AP is the set of possible space-time movement and geometric configuration conditions that must exist to

enable the affordance mechanisms to act and execute the capability (Michell, 2012). In the syringe injection case the affordance mechanism is the force from the doctor’s hands holding and pressing the plunger of a syringe to give a patient an injection. This force is transmitted along (a linear in this case) affordance path from the plunger into the drug fluid which is driven out into the patient (Michell and Roubtsova, 2014). The affordance transmission path for any capable action forms an affordance chain from the originating source of energy through the operating parts of the device.

We can identify the energy mechanism and the (affordance) path of the energy by observing the chain of actions to make the man-machine capability work (Michell and Roubtsova, 2014). At the point of executing the capability to inject for example the affordance chain, eg for injection will be a force transfer from hand to syringe, to drug fluid pressure, to patient. (See Michell and Roubtsova, 2014 for details). A specific man- machine combination will however have specific values and measures of specific factors such as force, velocity and energy, path topology and device volume and geometry limits. These must be within a certain range for the capability to occur. These capability affordance factors (CAF) such as the range of angles the syringe can be held at, or the minimum force to grip a syringe or maximum force possible before breaking it will characterise the capability and enables us to model and compare capabilities (Michell and Roubtsova, 2014). To study and compare the capabilities of additive vs subtractive manufacturing we therefore need to identify the a) energy transfer mechanisms of the two sets of machines and b) their resulting transmission or affordance path and c) the critical affordance factors and range of values that make it possible.

5 MECHANISM MODELLING

5.1 SM Mechanism Affordance Chain

The mechanism represents the difference in energy that causes an action to occur. In subtractive manufacturing the affordance mechanism is typically the application of force greater than the material strength to cut or remove the material to the desired shape of the final part. This process is typically achieved via electrical energy used to rotate electric motors to drive a rotary cutter and to move the cutter or part in a specific intersecting 3D space-time path. For example a lathe has a motor to rotate the billet

about an axis, whilst the cutter is stationary and moved at right angles to the work. In contrast in a milling machine the cutter is rotated via an electric motor and the billet is clamped to table that can move in x , y and z directions with three degrees of freedom.

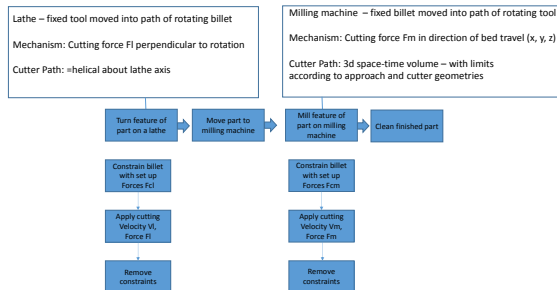


Figure 2: Example SM Mechanisms.

In subtractive manufacturing, unlike 3DP, there is no universal tool solution. A part, if complex, is produced with a series of machining actions on different machining centres, such as a lathe, mill, drill etc. Hence the affordance path comprises a series of cutting actions on separate machines, separated by moving the part between machine set up operations, where the billet is locked into the correct position and orientation for cutting. Each cutting operation comprises a relative simple mechanism or short affordance chain representing the application of a cutting force along a space time path in relation to a constrained billet as dictated by the machine and cutter geometry as in Figure 2.

5.2 Mechanism Driven CAF for Subtractive Manufacturing

Due to the forces involved in machining hard materials fixture constraints to hold the part being machined against the cutting forces may be required to be designed, positioned and applied, hence increasing the complexity and cost of setting up the additive manufacturing operation (Matthews, 2007).

There are upper and lower bounds in the forces required to remove unwanted material from a lump of metal or billet. For example there must be sufficient average torque T_m to cut the part but less than the maximum machine torque to damage material. A related factor is the tool spindle speed w , which again must be greater than a minimum value to ensure a smooth surface finish, but less than the critical value to cause burning or melting of the tool or part. In addition as tools are liable to be bent with high forces, the average cutting force F_m must be sufficient to cut smoothly with minimum tool deflection and less than the force required to cause part damage or failure such

to limit the average cutter deflection d_c will limit the capability of the tool to produce a quality part. Further the cutting force F_m must be less than the critical value necessary to cause significant tool wear (Blackenfelt, 2001).

The force required to cut a material and the material properties will dictate potential distortion especially in features of the manufactured product that have thin walls (Matthews, 2007). Further there is a limit to feature production of very fine parts. For example very thin cylinders or point features are not able to be machined due to these force limitations. Similarly the creation of a mesh from a solid billet would be extremely difficult, time consuming and very costly.

5.3 Mechanism for Additive Manufacturing

Additive manufacturing deposition mechanisms can vary widely from heated molten plastic to laser sintering (Norton, 2001). For capability comparison we will focus on the most typical mechanism used in the cheaper 3D printers – based on heated plastic deposition. In a typical 3D printer of this form there are 4 component mechanisms that form an affordance chain of interacting actions (Figure 3).

Firstly a material feed process is usually via electrically driven rollers applying a rotary constraining force to a fibre of the plastic working material which is dragged into an electrically driven heating element. This heats the plastic to a pliable temperature ϕ and viscosity within a range that enables the melted plastic to flow through the nozzle in a continuous stream according to the feed controller.

The plastic flows through the nozzle at a steady velocity V_p driven by the force F_f imparted to the plastic filament by the rotation of the pinch rollers.

For deposition path planning & control, the heater and nozzle are typically fixed to a framework of motor driven slides (similar to a 2D inkjet printer) that enables the precise positioning of a point or a line/area of the deposited material dictated by the optimised STL model of the structure driven by the CAD design model. The material deposited internal to the shape – the ‘infill’ is not solid but an STL lattice structure that can be 80% or more less than the full CAD solid model (Bassett et al., 2015).

Finally full x , y , z coverage of a three degree of freedom build space is achieved by lowering the x - y plane or build table in the z direction as successive layers are added. This table is often heated, or the build volume enclosed, to minimise warping of the

part which would otherwise occur with rapid cooling back to room temperature.

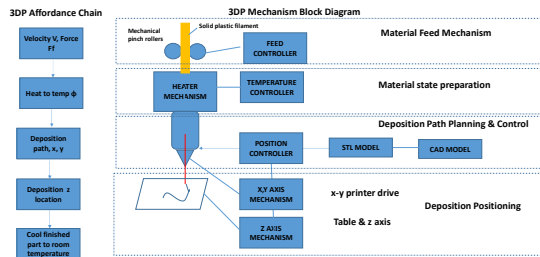


Figure 3: 3D Printer Key Components and Mechanism.

5.4 Mechanism Driven CAF for Additive Manufacturing

A capability affordance factors that limit and define whether the affordance and capability will occur for 3DP are based on the heat transfer and feed rate variables for the melted material deposition mechanism a sample of which is discussed below, but a wider range can be seen in (Choi and Samavedam, 2002).

Firstly the material melting temperature θ must be within a range that enables flow, but not burning of the PLA/ABS material. For example $175 < \theta_{pla} < 200$ degrees, $225 < \theta_{abs} < 230$ degrees (Basset et al, 2015). Secondly the build bed must be heated within a given temperature range to avoid warping as the part cools. For example $20 < \gamma_{pla} < 70$ degrees, and for ABS $105 < \gamma < 105$ degrees (Basset et al., 2015). The material filament feed rate V_f must be within a range consistent with smooth deposition to avoid gaps or bunching, or unwanted spreading of the material.

The material viscosity μ at this temperature must be sufficiently liquid to flow, but not too low to result in puddling and spreading. The x,y,z velocity of the nozzle travel V_{n-xyz} must also be in proportion to the material viscosity and flow rate and be above a minimum where material bunching will occur and less than a maximum that would cause gaps in the deposition. The deposition layer height h_d is important for part surface finish and fineness and is dependent on nozzle diameter, but must be above a minimum value that would unacceptably increase the time to produce the part (Basset et al., 2015). The infill structure density p must also be within a range of values that enables sufficiently rapid printing, but not too sparse to cause problems with structural integrity

Various combinations of these factors will affect the capability and mathematical models have been built to establish the range of control parameters and the physics and heat transfer involved (Ganeriwala

and Zohdi, 2014). For example the plastic feed rate must be related to the x, y, z velocity for a given material viscosity to avoid gaps or lumps in the deposited plastic. Feed rate and viscosity will affect the layer thickness, which in turn will affect surface accuracy as a series of stacked layers or 'stair step effect'. (Choi and Samavedam, 2002). The mechanism driven factors for AM and 3DP can be seen in Figure 5.

6 PATH MODELLING METHODS

To compare 3DP capability with traditional manufacturing we need to develop a conceptual model of the affordance path that takes into account the path geometry and the mechanism of the different methods. Both manufacturing methods involve the forming/sculpting of three dimensional solids, one by removing material from a solid block or billet of material by force and the other by deposition. Much research has been conducted into traditional manufacturing focusing on cutter path geometries (Feng and Cusiak, 1995) and the study of tool vibration effects (Blackenfelt, 2001). Other work has focused on identifying the optimum sequence and type of machining (Gupta et al, 1997) or machining process simulation (Blackenfelt, 2001) and the problem of modelling machining features (Tapie et al., 2012). However using a mathematical geometric approach to modelling these methods is unduly complex and unnecessary for our comparison purposes.

An alternative method that is appropriate to accommodate the massive variation in machine tools and machined or 3dp manufactured objects is to model the situation from a topological set theory perspective. Consideration of the problem suggests that for subtractive manufacturing we need to model the intersection of a polygonal volume with a cutter path with different degrees of freedom. The additive manufacturing (3DP) model can be considered as the intersection of a bound build volume space with a three degree of freedom deposition path.

Topological modelling defines spatial relationships for geometries that are preserved under rotation and scaling transformations (Egenhofer et al., 1994). Such modelling can be used to define topological relationships between two spatial objects (Borrmann et al., 2006). Based on the fact that boundaries and interiors have been identified as the crucial descriptions of polygonal intersections (Wagner, 1988), Egenhofer developed a point set model of topological spatial relations between

regions or areas (Egenhofer and Franzosa, 1991). The point set approach considered a set of points x and a set of points y with neighbouring and overlapping topologies defined by the set theory. Egenhofer defined the results of the intersections of the boundaries and interiors of two shapes in terms of non-empty $\neq \emptyset$ and empty sets \emptyset , resulting in nine feasible topological relations.

6.1 Three Dimensional Modelling

The 2D model was formally extended by specification of the interior, boundary and exterior point sets of spatial object by Egenhofer et al, who described the topological relations between two volumetric cells A and B . He considered A and B as arbitrary objects composed of sets of points (Egenhofer, 1994). These comprised three ‘object parts’; interior designated by $^\circ$, boundary designated by δ and exterior designated by $^-$ and the two point sets; $\{A^\circ, \delta A, A^-\}$ and $\{B^\circ, \delta B, B^-\}$.

The combinations of intersections between interior, boundary and exterior point sets result in the nine intersection model (9-IM).

$I =$		
interior a \cap interior b	interior a \cap boundary b	interior a \cap exterior b
boundary a \cap interior b	boundary a \cap boundary b	boundary a \cap exterior b
exterior a \cap interior b	exterior a \cap boundary b	exterior a \cap exterior b
$A^\circ \cap B^\circ$	$A^\circ \cap \delta B$	$A^\circ \cap B^-$
$\delta A \cap B^\circ$	$\delta A \cap \delta B$	$\delta A \cap B^-$
$A^- \cap B^\circ$	$A^- \cap \delta B$	$A^- \cap B^-$

Figure 4: The 9-IM Topological Intersection Model.

Borrmann et al’s work describes a generic intersection matrix for intersections of point, line, area and 3d body (Borrmann et al., 2006). This gives a number of intersection volumes that include; non-touching (disjoint), equal volumes, a volume containing and touching a volume and a volume totally within a volume (Borrmann and Rank, 2009). Whilst their work focused on modelling intersections of architectural buildings, as these are simply intersecting polygons, the same approach can be adapted to manufacturing of polygonal parts. This is equivalent of the intersection of the billet and the cutter path. For our purposes we only consider the body intersection which we can use to model the 3D billet, cutter path or deposition path.

Using the same principle, the ‘equal volumes’ topology can be used to describe the 3D printer situation, where the material deposition volume of a 3D printer is equal to the full 3D build space available. This represents a ‘total topological

relationship’ (Billen and Kurata, 2008), where any point, line, area or the full work volume can be reached with the three degrees of freedom of deposition of the 3D printer

Earlier in equation 3 we defined P_s as a part produced by subtractive manufacturing. As topological relations are indicated by the presence or absence of intersections in the 9-IM model, we can rewrite the model for AM and SM and apply the set theoretic approach to show which relationship holds.

For an additive manufacturing process the completed part P_a should be exactly the same as the part produced by subtractive manufacturing P_s . For P_a the topological path model is a function of the swept deposition volume P_d and the available build volume limits V . The finished part depends on the volume of deposited material defined by the set of points P_d in relation to the overall volume available to build in V .

$$P_s = P_a = V \cap P_d \tag{5}$$

Hence using the same 9-IM matrix $M(V, P_d)$ we can represent the possible topology for the 3D printer AM approach. Where A in the original 9M matrix represents the deposition path volume P_d from the deposition of melted plastic points in layers. B represents the available free space V to build in.

6.2 Capability Path Comparison

Using the above topologies the resulting manufactured part is described by the range of non-empty sets formed by the intersection of the four feasible relations in the 9-IM model and defines the gross path capability of each manufacturing approach as seen in figure 6. This clearly shows the versatility of the additive 3DP solution in its ability to deposit material in any part of the build volume. Additive manufacturing enables non empty sets for a part within a part represented by the ‘contains’ relation where an unused volume is completely contained by the build volume and by extension a separate part could also be present if deposition was stopped and restarted. It also holds for the touch relation where the internal void produced by AM touches an external boundary. In contrast for subtractive manufacturing all these options are empty sets and prohibited, apart from the intersection topology, which has limitations based on the capability affordance factors of accessibility and operation sequence discussed earlier.

However, despite its greater path flexibility 3DP has additional path conditions. The material deposition as a slug of melted plastic will vary with the curvature of the ‘tool’ or print head and thus build

orientation of the part will also affect final part quality (Nelaturi and Shapiro, 2015). In addition the cooling of the melted material in certain geometries may result in drooping and warping and hence certain features and overhangs may require additional build support structure to be added during construction to prevent this. In contrast the path driven capability affordance factors for subtractive manufacturing as discussed earlier are far more numerous, complex and onerous. See figures 5 and 7 for both sets of factors.

7 SUMMARY & CONCLUSIONS

7.1 Summary

We have analysed the perceived business and technical advantages of 3D printing and shown how these are derived by the mechanism and path limitations of the AM and SM approaches. We have compared the capability of 3D printing with traditional subtractive manufacturing using a capability affordance comparison method and a topological model. The model demonstrated that the mechanism of subtractive manufacturing is largely based on force application and that this leads to the requirements for extensive jigs and fixtures to hold the part in place. This compared to the heat transfer mechanism aided by a small filament force of 3D printing. The 3DP mechanism in contrast relies on gravity and the precise location of a point of material to constrain the finished part. However, the 3DP heat transfer mechanism produces a greater number of more difficult to control factors such as flow rate, viscosity etc as seen in Figure 5.

In contrast the topological model comparison emphasises how traditional subtractive manufacturing methods are limited by the path geometry of the cutters used and the need for cutter access. This is emphasised by the fact that a 3D printer with three degrees of freedom of point material deposition has a ‘total topological relationship’,

compared with the ‘partial topological relationship’ of subtractive machining methods. SM is further limited by the need, especially for complex parts, to carefully consider the order of subtractive operations, demonstrated by the larger range of geometric critical affordance factors than 3DP seen in Figure 7.

However, whilst additive manufacturing has a far greater capability to produce complex and intricate parts, the machine parameters involved with the deposition approach can be more difficult to get right to ensure a quality part compared with tried and trusted subtractive force driven mechanisms where much more is known about the kinematics and thermodynamics involved ensuring consistent quality part production.

7.2 Conclusion

We have explored the key business and technology benefits of 3DP additive manufacturing compared with traditional subtractive manufacturing. We applied the capability affordance model (CAM) defining capability as an energy mechanism operating through a space time path. We have shown via topology analysis that the key capability difference and advantage for 3DP is in its superior space time affordance path due to the three degrees of freedom to build from a point to enable construction of complex single part objects and meshes, parts within parts and very fine constructions. This is superior to the limited topology available to additive manufacturing due to the need for cutter access and geometry limitations. However the force focused mechanism of additive manufacturing does enable more consistent surface quality and durability compared to the heat flow focused mechanism of 3DP with its more complex capability affordance factors that lead to this variation. However, the low cost, ease of use and design friendly format of 3DP enables significant benefits to small and medium manufacturers in producing innovative product capabilities over traditional methods.

Subtractive Manufacturing (SM) Capability Affordance Factors AFFORDANCE MECHANISM DRIVEN		3DP Additive Manufacturing (AM) Capability Affordance Factors AFFORDANCE MECHANISM DRIVEN	
SCM1	sufficient to cut material < Average machine Torque T_m < torque to damage material	ACM1	$175 < \theta_{pla}$ material deposition temperature $\theta_{pla} < 200$ degrees
SCM2	speed to produce rough surface < tool spindle speed w < tool/part burning or melting speed	ACM2	Build bed temperature For example $20 < \gamma_{PLA} < 70$ degrees, 105
SCM3	sufficient to cut < average tool force F_m < force required for surface damage due to tool deflection	ACM3	velocity for gaps to occur < material filament feed rate velocity V_f < velocity for unwanted spreading
SCM4	sufficient to cut < F_m < distortion force for thin wall sections	ACM4	minimum flow value < material viscosity μ < viscosity for puddling and spreading
SCM5	sufficient to cut < F_m < force required to cause significant tool wear	ACM5	minimum for speed requirements < deposition layer height h_d < surface finish fineness value
		ACM6	velocity for bunching of deposition material < The 3D nozzle velocity V_n -xyz < velocity for gaps in deposition
		ACM7	density for structural issues < infill structure density p < density for maximum print time
		ACM8	material specific heat capacity < critical value for failure (burning/part damage)

Figure 5: Affordance Mechanism Comparison.

		PATH CAPABILITY Subtractive mfg			PATH CAPABILITY Additive mfg			
	EQUAL	-∅	∅	∅	no	lack of feasible cutter path to access all points on a billet (limited by cutter size and geometry and kinematics)	yes	ability to deposit material anywhere in the workspace
	CONTAINS a contains b	-∅	-∅	-∅	no	lack of feasible cutter path completely inside the billet	yes	ability to deposit material to completely envelope part of the
	INTERSECT	-∅	-∅	-∅	yes	feasible cutter paths to produce intersection (limited by cutter size and geometry and kinematics)	yes	ability to deposit material in part of the workspace
	TOUCH	-∅	-∅	-∅	no	lack of feasible cutter path completely inside the billet to a single point on the boundary	yes	ability to deposit material to completely envelope the workspace except for an

Figure 6: Comparison of Capability Paths.

Subtractive Manufacturing (SM) Capability Affordance Factors AFFORDANCE PATH DRIVEN		3DP Additive Manufacturing (AM) Capability Affordance Factors AFFORDANCE PATH DRIVEN	
SCP1	Order of Processing enables all manufacturing features to be exist(OOP)	ACP1	part orientation < critical path orientation geometry
SCP2	Accessibility is within access constraints - (AC)	ACP2	geometry to support requirements for overhanging parts
SCP3	Datum Dependency - (DD) conditions are met		
SCP4	Approachability – (AP) conditions are met		
	Cutter Access Path limitations (CPL)		
SCP5	CPL1: The curvature d_{cur} of the tool < min curvature of concave edges		
SCP6	CPL2: tool the diameter of the passage d_{pas} < tool diameter		
SCP7	CPL3: Concave corner diameter (d_{cor}) formed by the cutter path > tool diameter		
SCP8	CPL4: cutting trajectory d_{tra} must be sufficient to enable the cutter to avoid cutting into the part		
SCP9	CPL5: The max diameter of tool d_{max} < smallest diameter feature (min d_{min})		

Figure 7: Affordance Path Driven Capability Affordance Factors.

REFERENCES

- Bassett, K., Carriveau, R., & Ting, D. K. (2015). 3D printed wind turbines part 1: Design considerations and rapid manufacture potential. *Sustainable Energy Technologies and Assessments*, 11, 186-193.
- Berman, B. (2012). 3-D printing: The new industrial revolution. *Business horizons*, 55(2), 155-162.
- Blackenfelt, M. (2001). Managing complexity by product modularisation
- Billen, R., & Kurata, Y. (2008). Refining Topological Relations between Regions Considering Their Shapes. In *Geographic Information Science* (pp. 20-37). Springer Berlin Heidelberg.
- Borrmann, A., Van Treeck, C., & Rank, E. (2006, June). Towards a 3D spatial query language for building information models. In *Proc. Joint Int. Conf. of Computing and Decision Making in Civil and Building Engineering (ICCCBE-XI)* (Vol. 2).
- Borrmann, A., & Rank, E. (2009). Topological analysis of 3D building models using a spatial query language. *Advanced Engineering Informatics*, 23(4), 370-385.
- Choi, S. H., & Samavedam, S. (2002). Modelling and optimisation of rapid prototyping. *Computers in industry*, 47(1), 39-53.
- Conner, B. P., Manogharan, G. P., Martof, A. N., Rodomsky, L. M., Rodomsky, C. M., Jordan, D. C., & Limperos, J. W. (2014). Making sense of 3-D printing: Creating a map of additive manufacturing products and services. *Additive Manufacturing*, 1, 64-76.
- D'aveni, R. A. (2013). 3-D printing will change the world. *Harvard business review*, 91(3), 34-35.
- Depoorter, B. (2013). Intellectual Property Infringements & 3D Printing: Decentralized Piracy. *Hastings LJ*, 65, 1483.
- Dobrovski, E. L., Verlinden, J. C., & Geraedts, J. M. P. (2011) From factory to replicator
- Egenhofer, M. J., & Franzosa, R. D. (1991). Point-set topological spatial relations. *International Journal of Geographical Information System*, 5(2), 161-174.
- Egenhofer, M. J., Clementini, E., & Di Felice, P. (1994). Topological relations between regions with holes. *International Journal of Geographical Information Science*, 8(2), 129-142.
- Egenhofer, M. J. (1994). Deriving the composition of binary topological relations. *Journal of Visual Languages & Computing*, 5(2), 133-149.
- Feng, C. X., & Kusiak, A. (1995). Constraint-based design of parts. *Computer-Aided Design*, 27(5), 343-352.
- Ganeriwala, R., & Zohdi, T. I. (2014). Multiphysics modeling and simulation of selective laser sintering manufacturing processes. *Procedia CIRP*, 14, 299-304.
- Gibson, J.: *The Ecological Approach to Visual Perception*. Houghton Mifflin Company, Boston (1979) *California Management Review*.
- Gupta, Satyandra K., William C. Regli, Diganta Das, and Dana S. Nau. "Automated manufacturability analysis: a survey." *Research in Engineering Design* 9, no. 3 (1997): 168-190.
- Hoy, M. B. (2013). 3D printing: making things at the library. *Medical reference services quarterly*, 32(1), 93-99.
- Matthews, J. A. (2007). A constraint-based approach for assessing the capabilities of existing designs to handle product variation (Doctoral dissertation, University Library).
- Michell, V.A.(2011) A Focused Approach to Business Capability. First International Symposium on Business Modelling and Software Design – BMSD 2011, Sofia, Bulgaria, pp. 105–113.
- Michell V. (2012) The Capability Affordance Model: Comparing Medical Capabilities. In: B. Shishkov (Ed.) *Business Modeling and Software Design – BMSD'12 Revised Selected Papers*, Springer-Verlag – Lecture Notes in Business Information Processing, Berlin-Heidelberg
- Michell, V., & Roubtsova, E. (2014). Modelling Capability and Affordance as Properties of Human/Machine Resource Systems. In *Proceedings of the 4th International Symposium on Business Modeling and Software Design, BMSD*
- Nair 2014 democratising-manufacturing-3d-printing-is-set-to-change-the-startup-landscape. Available at <http://www.thestar.com.my/business/sme/2014/03/26/democratising-manufacturing-3d-printing-is-set-to-change-the-startup-landscape/> [Accessed 04 May 2016]
- Nelaturi, S., & Shapiro, V. (2015). Representation and analysis of additively manufactured parts. *Computer-Aided Design*, 67(C), 13-23.
- Norton, A. (2001). Utilising Rapid Product Development and Late Customisation Methodologies within Manufacturing SMEs. PDF article to link: <https://www.google.com/#q=Rapid+Product+Development+and+Late+Customisation>, 15-16.
- Pham, D. T., & Gault, R. S. (1998). A comparison of rapid prototyping technologies. *International Journal of machine tools and manufacture*, 38(10), 1257-1287.
- Rosen, D. W. (2007). Computer-aided design for additive manufacturing of cellular structures. *Computer-Aided Design and Applications*, 4(5), 585-594.
- Tapie, L., Mawussi, B., & Bernard, A. (2012). Topological model for machining of parts with complex shapes. *Computers in Industry*, 63(5), 528-541.
- Turvey, M.T. (1992). Affordances and Prospective Control: An Outline of the Ontology. *Ecological Psychology* 4(3), 173–187.

A Multi-criteria Evaluation Framework for Selecting Sensitive Business Processes Modeling Formalism

Mariam Ben Hassen, Mohamed Turki and Faïez Gargouri
ISIMS, MIRACL Laboratory, University of Sfax, P.O. Box 242, 3021 Sfax, Tunisia
{mariembenhassen, mohamed_turki}@yahoo.fr, faiez.gargouri@isims.rnu.tn

Keywords: Knowledge Management, Sensitive Business Process, Business Process Modeling, Business Process Meta-model.

Abstract: Sensitive business process (SBP) modeling has become primary concern for any successful organization to improve the management of their individual and collective crucial knowledge on which it is necessary to capitalize. This paper presents a multi-perspective evaluation framework for assessing the expressiveness of current widely used BPM formalisms, in order to select the most suitable for the SBP representation and improve the identification of crucial knowledge that is mobilized by these processes. Furthermore, the result of the evaluation led us to justify the choice of the better one positioned nowadays, the standard BPMN 2.0. Besides, we have illustrated the practical applicability of this notation on a medical process in the context of the association of protection of the motor disabled people of Sfax-Tunisia (ASHMS).

1 INTRODUCTION

Today's organizations has been characterized by collaborative, highly dynamic, complex and knowledge-intensive actions. In order to improve competitive advantage, they have been increasingly conscious of the necessity to formalize and capitalize knowledge produced and mobilized by their business processes (BPs). According to this view, business process modeling (BPM) has become crucial concern for successful organizations to improve the identification, acquisition, storage, dissemination, sharing, creation and (re) use of their individual and organizational knowledge.

Considering the large amount of knowledge to be preserved and enhanced, such organizations must first identify and model the SBPs which are likely to mobilize crucial knowledge on which it is necessary to capitalize. In fact, the more organization's BPs are sensitive, the more they can mobilize crucial knowledge. Few existing research on Knowledge Management (KM)-BPM focusing on the identification, analysis and modeling of SBPs in order to localize and identify the crucial knowledge. We quote: the Global Analysis METHodology (GAMETH) proposed by Grundstein (2000), the identifying crucial knowledge methodology (Saad et al., 2009) and the Sensitive Organization's Process

Identification Methodology (Turki et al., 2014a). However, these methods do not explicitly and conveniently address the critical operation of « SBPs modeling ». A SBP typically lacks a description and a representation that allow to explicit the rich semantics embedded into a SBP. So, the specification of a precise conceptualization, with a subjacent representation notation, that explicitly and adequately integrate the knowledge dimension within their actions and other relevant SBP aspects, is still an open issue. In fact, a SBP has its own characteristics that distinguish them from classical BPs. In fact, a SBP commonly mobilizes a high number of critical activities with very specific knowledge « crucial knowledge» (tacit and explicit). It presents a diversity of knowledge sources and possesses a high degree of dynamism in the objectives' change and high complexity.

Some conventional graphical BPM formalisms, include, amongst others, Event Driven Process Chain (EPC) (Korheer and List, 2006), Business Process Modeling Notation (BPMN 2.0) (OMG, 2011a), Unified Modeling Language (UML 2.0) activity diagram (OMG, 2011b), Specification Language (PSL) (Schlenoff et al., 2000), Process Business Process Modeling Ontology (BPMO) (Cabral et al., 2009) and Role Activity Diagram (RAD) (Weidong and Weihui, 2008), have been adapted to allow the representation of the intrinsic

elements of knowledge within BPs. But, these languages/notations do not include all the required features to describe a SBP. In addition, the literature shows a set of approaches dedicated to knowledge highly intensive processes (KIPs) representation, originate from the knowledge modeling context, including the Business Process Knowledge Method (BPKM) (Papavassiliou and Mentzas, 2003), DECOR (Abecker, 2001), CommonKADS (Schreiber et al., 2002), Knowledge Transfer Agent (KTA) Modeling Method (Strohmaier et al., 2007), PROMOTE (Woitsch and Karagiannis, 2005), the work of Donadel (Donadel, 2007), DCR Graphs (Hildebrandt and Mukkamala, 2010), Knowledge Modeling Description Language (KMDL 2.2) (Gronau et al., 2005) (Arbeitsbericht, 2009), GPO-WM (Heisig, 2006), Oliveira's methodology (Oliveira, 2009), and the Notation for Knowledge-Intensive Processes (NKIP) (Netto et al., 2013), etc. However, none of these proposals, as shown in (Ben Hassen et al., 2015a), adequately addresses all the relevant SBP elements.

In order to address existing limitations and improve the SBP representation, we proposed, in previous work (Ben Hassen et al., 2015a) (Ben Hassen et al., 2015b), the Business Process Meta-Model for Knowledge Identification (BPM4KI) BPM4KI comprises concepts from several perspectives that are crucial for a complete understanding, characterization and representation of a SBP, namely the functional perspective, the organizational perspective, the behavioral perspective, the informational perspective, the intentional perspective and the knowledge perspective. The generic meta-model we have developed is semantically rich and well founded on COOP, a core ontology of organization's processes proposed by Turki et al. (2014b) which is useful to characterize the concepts useful for the analysis and identification of SBPs. Furthermore, BPM4KI serves as a comprehensive evaluation framework of the expressiveness and adequacy of current widely-used BPM formalisms, to check their suitability to cover all the relevant elements of a SBP. Precisely, the (objective) evaluation facilitates selecting and justifying the most appropriate BPM formalism for the representation of SBP taking its semantic dimensions into account.

The overall goal of the present work is to carry out an evaluation of which BPM4KI elements are potentially supported by the above-mentioned language meta-models. Besides, it presents a practical example using the best evaluated formalism. Furthermore, it points alternatives for representing elements that not adequately addressed yet.

The remainder of the paper is structured as follows. Section 2 presents the core concepts that describe Sensitive Business Process and related work about modeling SBP. Section 3 presents the main characteristics of current formalisms for BPM, evaluates their suitability to support the representation of all relevant SBP elements. Section 4 presents a practical example. Section 5 concludes the paper and underlines some future research topics.

2 SENSITIVE BUSINESS PROCESSES

2.1 SBP Fundamentals

According to Ben Hassen et al. (2015b), a SBP represents the core process of organization which constitutes the heart of the organization's activities. It is commonly mobilizes very specific knowledge «crucial knowledge» (i.e. the most valuable/important knowledge on which it is necessary to capitalize). It includes a high number of critical activities which mobilizes and produces different types of knowledge: (i) imperfect individual and collective knowledge (tacit and/or explicit) (i.e. missing, poorly mastered, incomplete, uncertain, etc.) which are necessary for solving critical determining problems; (ii) a great amount of heterogeneous knowledge recorded on diverse knowledge sources (dispersed and sometimes lacking accessibility); (iii) expertise and/or rare knowledge held by a very small number of experts; flexible knowledge owned by experts; (iv) very important tacit organizational knowledge (like competences, abilities and practical experiences).

Moreover, it contains activities that valorize the acquisition, storage, dissemination, sharing, and creation and (re) use of individual and organizational (tacit and explicit) knowledge, in the sense that it mobilizes a large diversity of knowledge sources consigning a great amount of very important heterogeneous knowledge. Its execution involves a large number of business domains/competencies (in terms of internal and external organization unit/agents operating in the BP), having distinct experience and expertise levels. Furthermore, it include a high number of organizational collaborative activities that mobilize, exchange, share and generate new individual and collective knowledge that is created by dynamic conversion of existing ones in the process in order to achieve organizational objectives. So, it depends on

knowledge flows and transfer of data, information and knowledge objects between communicating process participants. Other typical characteristics of SBPs presented in Ben Hassen et al. (2015b) includes: (i) A SBP is unstructured or semi-structured. Yet, a flexible process typically contains a very dynamic and unpredictable control-flow, comprising complex activities (individual and /or collective) that may frequently change over time or at design-and run-time. The process agents (e.g. experts) is often not able to predetermine the overall process structure in terms of the activities to be executed and their ordering, the data and knowledge sources to be exploited and the roles and resources required for process progression and completion. (ii) It is driven by constraints and rules. Indeed, process participants may be influenced by or may have to comply with constraints and rules that drive organizational actions performance and decision making. (iii) It possesses a high degree of dynamism in the objectives' change associated to it, essentially, in decision making context. The change of organizational objective leads to a new organizational distal intention (which is necessary to control the SBP) and influences experts' decision making. (iv) Its contribution to reach strategic objectives of the organization is very important. Also, their realization duration and cost are important.

According to above mentioned, representing and organizing the knowledge involved in SBPs is very complex, especially when applying traditional approaches. However, it is difficult to find out an approach/formalism that addresses all or at least most of these characteristics in the representation of a SBP model. Nevertheless, the Object Management Group (OMG, 2011a) states that, in addition to underlining the concepts inherent to a domain, a notation enhances the clarity of the models and allows the ability of communicating the concepts uniformly. The selection and adoption of a suitable BPM formalism for representing SBP models is still an open issue, allowing the knowledge mobilized and generated by the BP instances to be located, identified, modeled, stored and reused. In this context, several BPM approaches and notations are found in literature as likely to represent SBP.

2.2 Related Work: SBP Modeling Approaches

Although there is abundance of BPM formalisms and despite their diversity, only a few were applicable for SBP modeling. Some traditional workflows/BPM formalisms that are widely-

followed in current research and practice scenarios (such as BPMN, EPC, UML AD), have been adapted to allow the representation of the intrinsic elements of knowledge within BPs. However, they were not meant for SBPs, since they focus on the representation of "deterministic" process, composed by a well- structured control flow among its activities, low uncertainty and complexity (that is, the existence of few and pre-defined exceptions). Besides, these notations can be used to implicitly identify certain issues related to knowledge flows, such as the information sources that are required, generated, or modified by an activity.

Moreover, some authors have attempted to develop approaches for the representation of processes with high knowledge intensity (KIP) (Gronau et al., 2005) (called also Process-oriented knowledge modeling approaches) where basic phenomenon is knowledge. In these processes, the principal success factor is adequate modeling of knowledge conversions. It is noteworthy that SBP shares many common characteristics with KIP approaches. In fact, KIPs are processes whose conduct and execution are heavily dependent on knowledge workers performing various interconnected knowledge intensive decision making tasks. KIPs are genuinely knowledge, information and data centric and require substantial flexibility at design- and run-time. These approaches that focus on KM within the BP level have not been widely adopted by organizations and are still very incipient. Also, they have limited capabilities, in the sense that they do not conveniently include process perspective, as well as they do not provide an opportunity to clearly distinguish between data, information and knowledge.

The CommonKADS (Schreiber et al., 2002) approach focuses on knowledge representation. Various stages of modeling attempt to establish a structured approach so that knowledge can be managed with the support of technical and engineering tools. Three basic points characterize these demands: the details of the skills involved in process execution, the representation of the processes through artifacts and semantic analysis, and the opportunities for improvement regarding the process and use of knowledge. The BPKM-Business Process Knowledge Method (Papavassiliou and Mentzas, 2003) provides a methodological guidance for the implementation of BP-oriented KM. It presents a meta-model for integrating BPM aspects with KM. This meta-model transcribes the four perspectives of a workflow: task, organizational, logical and data. It was extended to

include KM tasks that support BPs represented by the elements: knowledge management task, knowledge object and knowledge archive. Two other approaches of knowledge representation are the Knowledge Transfer Agent (KTA) Modeling Method (Strohmaier et al., 2007) and the DECOR approach (Abecker, 2001). The first describes how to create knowledge transferring models. The method consists of modeling and analyzing in three distinct levels of detail. The DECOR Project delivers context-sensitive organizational knowledge and has its focus in representing processes knowledge across diagrams embedded in organizational memory. It aims to structure the BP, the dynamic context, contextual information and the representations of memories embedded in the production process. In the method proposed by Donadel (2007) aims to support the management of knowledge resources related to BPs. The organizational value chain is mapped and the knowledge aspects that may influence the organizational processes are represented. The aforesaid knowledge oriented approaches do not explicitly differentiate between tacit and explicit which is relevant in SBPs due to, for instance, the high degree of tacit knowledge developed and exchanged among agents through inter-organizational collaboration. And most of them do not provide special attention to the graphical notation for BP representation.

Furthermore, knowledge is modeled using another specific knowledge modeling notations (e.g., KMDL, GPO-WM, Oliveira) and only few of them include process perspective (e.g., PROMOTE, RAD)). The Knowledge Modeling Description Language (KMDL) (Gronau et al., 2005) (Arbeitsbericht, 2009) formalizes KIPs with a focus on certain knowledge-specific characteristics in order to identify process improvements in these processes. It represents both tacit and explicit knowledge of the process. Thus, the different possibilities of knowledge conversion can be modeled and the flow of knowledge between actors is depicted. However, this notation does not distinguish between data and information, and does not address the representation of artefacts and dynamic aspects of BP and modeling agents. Besides, it is hard to understand and to apply for the purpose of facilitating the involvement of modeling participants. Method for integration of KM into BPs (GPO-WM) (Heisig, 2006) describes and evaluates the current state of handling core knowledge domains, to gather improvement ideas for systematic knowledge handling and to integrate selected KM

methods and tools into existing BPs. The notation does not allow the modeling of knowledge conversions. The abovementioned proposals focus on storing and sharing knowledge. Thus, they lack the ability to model in an adequate manner the decisions, actions and measures, which are causing a sequence of processes. Most of these methods are convenient only for knowledge management experts and require additional training for non-experts. The method for integrated modeling of BPs and knowledge flow based on a Role Activity diagram (RAD) (Weidong and Weihui, 2008) provides integration of BPs and knowledge flow and helps KM build on existing process management efforts. This method does not differentiate between tacit and explicit knowledge and does not present different types of knowledge conversion—that are relevant in SBP. Also, it does not present and separate data and information from knowledge. Supulniece et al. (2010) argued an extension of BP models with the knowledge dimension in order to take advantage of some opportunities such as identifying, planning and managing required knowledge for the role that participates in a particular activity; evaluate the amount of lost knowledge if a person would leave the organization; improve understanding about the knowledge usefulness, validity and relevance for particular activities; enable competence requirements management and proactive training. They extended BPMN incorporating concepts defined by KMDL (Gronau et al., 2005), where three different objects: knowledge objects, information objects and data objects were used. However, the proposed approach does not present knowledge flow between process participants; it lacks information about the knowledge structure; it does not integrate and separate the different knowledge types (like experience, basic knowledge, general knowledge) and it does not explicitly represent the tacit knowledge that is owned by a particular person. Recently, Netto et al. (2013) proposed KIPN, a notation for building KIPs graphical model that promotes the cognitively-effective understanding of this process. KIPN covers all characteristics defined by the knowledge-intensive processes ontology (KIPO) (França et al., 2012). It comprises a set of diagrams to represent the main dimensions within a KIP: the KIP, socialization, decision and good diagrams. In KIPN, activities are detailed through socializations. The agents interact and collaborate, contributing to the creation and acquisition of knowledge. Agents' contribution is represented by innovation, intention, belief, desire, feeling, experience and mental image elements, that are

difficult to be predicted and then modeled. The notation is able represent tacit knowledge through informal exchange and mental image elements, but it still does not capture explicitly the knowledge conversion. Moreover, NKIP is very incipient, hard to understand, not yet used and applicable for KIP modeling in current research and practice scenarios and not adopted by any available modeling tools.

Despite it mobilizes crucial knowledge within an organization and their key role for organizational KM, existing BPM approaches/notations have shortcomings concerning their ability to explicitly incorporate the knowledge dimension within BPs models as well as relevant issues at the intersection of KM and BPM. None of those proposals conveniently includes or addresses all or at least most of the SBPs important characteristics presented previously (critical activities (individual and/or collective), intensive acquisition, sharing, storage and (re)use of knowledge in challenging activities, large number of agents (external and internal) who have various business domains and different knowledge levels, high degree of tacit knowledge mobilized and exchanged among many experts, diversity of information and knowledge sources involved, high degree of collaboration (intra/inter-organizational) among agents/experts, dynamic conversion of knowledge, flexibility and dynamic aspects, deliberate actions, the influence of (distal) intentions in achieving objective and decision making, etc.). This leads to ambiguity and misunderstanding of the developed SBPs models.

2.3 SBPs Specification

In our previous research (Ben Hassen et al., 2015a), we have proposed a semantically rich conceptualization for describing a SBP organized in a meta-model, the Business Process Meta-model for Knowledge Identification (BPM4KI), which integrates all aforementioned perspectives. This meta-model intends to develop a rich and expressive graphical representation of SBPs in order to improve the localization and identification of crucial knowledge. BPM4KI is a well-founded meta-model whose concepts and relationships are semantically enriched by the core ontology organization's processes (COOP) (Turki et al., 2014b). BPM4KI covers all relevant aspects of BPM and KM within a SBP, and is composed by six perspectives:

(i) Functional Perspective, represents the BP elements which are being performed. The main concept that reflects this dimension is *Action*. It includes: *Individual Action*, *Collective Action*,

Action of Organization, *Inter Organizational Action*, *Organizational Action /Activity*, *Organizational Individual Action*, *Task*, *Organizational Unit Action*, *Organizational Sub Process*, *Organizational Critical Activity*, *Organizational Intensive Activity* and *Organizational Collaborative Activity*.

- (ii) Organizational Perspective, represents the different participants (the organizational resources) invoked in the execution of process elements as well as their affiliation. It display the process flows between different organizations and participants involved. The basic element of this perspective is *Agentive Entity* and includes: *Collective*, *Organization*, *Organization Unit*, *Human*, *Expert*, *Internal Agent*, and *External Agent*.
- (iii) Behavioral perspective, basically presents the logical sequence of elements to be executed in a BP. It includes synchronization, sequence, feedback-loop, complex decision requirements, in-and output criteria, etc. The basic element of this perspective is *Control Object* (such as control flow elements, pre-conditions, post-conditions, triggers, performance indicators, constraints, business rules, etc.).
- (iv) Informational perspective, describes the informational entities (such as data, artefacts, products and objects) which are generated, consumed, or exchanged within a process or an activity. It also includes both their structure and the relationships among them. The following concepts are related to this dimension: *Resource*, *Material Resource* (like informational and software resources), *Physical Knowledge Support*, *Event*, *Contingency*, *Input Object* (like data and information), *Output Object* (as data, information, services and results) and *Collaboration Protocol*.
- (v) Intentional perspective, provides an overview perspective of the process and captures important BP context information. It describes major BP characteristics and addresses the intentional information (such as objective, strategies, quality characteristics, metrics, measurement units, the deliverables, the process type and the customer), in order to ensure the BP flexibility. It comprises: *Intention*, *Objective*, *Distal Intention*, *Collective Intention*, *Collective Distal Intention*, *Organizational Distal Intention*, *Objective*, *Individual Objective*, *Collective*

Objective, Organizational Objective, Strategic Objective, Operational Objective, Deliberate Action, Culminated Process, Output Object (deliverables), Control Object (e.g., performance measures, constraints, business rules, etc.), Client, Sensitive Business Process, Knowledge Intensive Process, Inter Organizational Process, Internal Process, External Process, Partial External Process, Inter Functional Process, Core Process, Management Process, Strategic Process, Operational Process, etc. (which are some process types).

- (vi) Knowledge perspective, provides an overview perspective of the organizational and individual knowledge mobilized and created by an BP/organization as well as the knowledge flow proceeding within and between BPs/organizations. It addresses all relevant aspects related to KM (collection, organization, storage, transfer, sharing, creation and reuse among process participants). This vantage presents: *Knowledge (as an Immaterial Resource), Tacit Knowledge, Individual Tacit Knowledge, Collective Tacit Knowledge, Explicit Knowledge, Individual Explicit Knowledge, Collective Explicit Knowledge, Expert and Physical Knowledge Support*. It should be noted that some concepts are shared by different perspectives. For instance, *Collaborative Organizational Activity* and *Critical Organizational Activity* belong to all perspectives.

Nevertheless, BPM4KI does not provide a specific graphical notation for representing SBP. Although BPM4KI does not address the problem of representing SBP graphically, it opens a way to explore the potential of traditional BPM formalisms for it, as well as the usage of the specific process-oriented knowledge modeling/ KIP approaches. In the following section, we discuss the usage of BPM4KI concepts (which represent our evaluation framework) as a basis to model SBPs graphically.

3 A MULTI-PERSPECTIVE FRAMEWORK FOR COMPARATIVE ANALYSIS OF SBPS REPRESENTATION FORMALISMS

Based on the potential of BPM4KI to portray the

essential features of SBP, this section presents a comparative analysis of different BPM formalisms to represent SBPs. Precisely, in this research work, BPM4KI acts as a multidimensional evaluation framework for assessing the suitability of six selected BPM formalisms to cover all or at least most relevant elements of a SBP. We consider guiding and justifying the choice of the most suitable formalism for SBPs representation to characterize and improve the knowledge localization.

Before we present our evaluation framework for SBP representation, we will briefly refer to some related work about BPM languages (i.e. comparison and analysis) available in the field of meta-modeling and ontology. Many frameworks ((Söderström et al., 2002); (Lin et al., 2002); (Mendling et al., 2004); (List and Korherr, 2006)) have been proposed for evaluating the suitability of some BPM languages for specific purpose, according to generic meta-models. Most of them only focus on some aspects of BPM languages. Besides, the BWW (Bunge-Wand-Weber) ontological framework (Wand and Weber., 1990) has been widely used for assessing the ontological completeness and clarity of BPM languages, include (Rosemann et al., 2006); (Recker et al., 2009); (Penicina, 2013) and (Prezel et al., 2010). Furthermore, several works addressing the integration of KM into BPs, incorporating the knowledge into BP models. França et al. (2012) proposed KIPO, a formal meta-model/ontology that highlights the key concepts and relationships characterizing KIPs and used it as a reference for evaluating the adequacy of some existing BPM languages to represent each concept. However, this meta-model is not well adapted to represent SBPs. Sultanow et al., (2012) created a systematic comparison of thirteen selected methods based on a multidimensional framework to summarize the differences, also the most suitable situation for using each method. However, this framework do not consistently support SBP model requirements and concepts. Therefore, considering existing research in the KM-BPM domain, the knowledge dimension (i.e. the knowledge required to perform activities, the knowledge created as a result of BP activities, the sources of knowledge and their localization, the explicit knowledge, the tacit knowledge, individual and collective dimension of knowledge/activities, the knowledge flows between sources and activities, the different opportunities of knowledge conversion, etc.) needed for BPM is not explicitly represented, integrated and implemented in BP meta-models.

Hence, a comprehensive evaluation framework of the representational capabilities of current BPM

formalisms for SBPs is missing.

For discussion purposes, in this paper we take the constructs from BPM4KI as a relevant set of elements that are required to precisely represent a SBP, and evaluated some existing formalisms, which are based on different fields, to verify their suitability to cover and represent each concept. The multi-dimensional evaluation provides not only a useful framework to summarize the advantages and limitations of each formalism, but also select the most suitable positioned nowadays for SBP modeling, in order to localize the knowledge mobilized and created by these processes, which may be crucial. The evaluated representation languages were UML AD, BPMN 2.0, eEPC (which are adopted by many available modeling tools in current organizations), PROMOTE, KMDL 2.2 and Oliveira's methodology.

3.1 The BPM Formalisms - An Overview

In this section, we describe the BPM formalisms which have been chosen for evaluation. Some are process oriented and some are knowledge oriented. They represent the most frequently studied BPM formalisms in scientific/professional literature and practice scenarios.

UML 2.0 Activity Diagram (UML AD): UML AD (OMG, 2011b) in the behavior category are typically used for BPM. It is mainly and originally for modeling IT systems. UML AD is a semi-formal language with the following basic graphical notations: initial node and activity final node, activity, flow/edge, fork and join, decision and merge, partition/swimlane. This diagram is more expressive for modeling data flows inside information system and is less suitable for BPM.

Extended Event Driven Process Chain (eEPC): EPC (Scheer, 2000) is a semi-formal graphical modeling language for modeling, analyzing, and redesigning BPs, easily understood and used by business people. The basic notations include events, functions and connectors. It emphasizes more on the operational/functional and control perspectives than data transaction perspective. The basic version of EPC was supplemented by other constructs (organizational unit, position, information object, service object and application, resulting in the extended EPC (eEPC), intended to supplement process models with organizational structure and data flow. In eEPC, knowledge is represented by two object types, knowledge category and documented knowledge, and can be model by two

model types, knowledge structure diagram and knowledge map. In the first diagram, knowledge categories can be organized into subgroups based on their content. While the second depicts the distribution of various knowledge categories within an organization.

Business Process Modeling Notation (BPMN):

BPMN 2.0 (OMG, 2011a) represents the most popular and widely acceptable graphical notation to represent BPs, understandable by all business stakeholders, one that has now been ratified by the OMG as a BPM standard. It divides process knowledge into broadly five categories: flow-objects, connectors, artifacts, swimlanes and data. BPMN is initiated as a standard BPM language for conventional business, B2B and services process modeling. Hence BPMN has the capabilities of handling B2B business process concepts, such as public and private processes and choreographies, as well as advanced modeling concepts, such as exception handling and transaction compensation in addition to the traditional BP. The Collaboration and Choreography Diagrams allow modeling interaction among process' actors (between business partners, or different departments in a same company, members of a teams or even single workers and software systems), who exchange messages, while performing their tasks to reach a common objective.

PROMOTE: The PROMOTE (Woitsch and Karagiannis, 2005) integrates strategic planning with the evaluation of KM and BP management and defines KM requirements on the basis of business needs. It captures, models and evaluates the knowledge in enterprises and KIPs. It provides three diagram types: a knowledge diagram, a knowledge application diagram and an evaluation diagram. This notation can specify the knowledge conversion types. But, it does not explicitly separate tacit and explicit knowledge.

Knowledge Modeling and Description Language

(KMDL): is a semi-formal modeling method for the detection, visualization, analysis and evaluation of BPs and knowledge flows (Gronau et al., 2005). It increases the transparency of the existing knowledge in enterprises and optimizes the process of KIPs. This notation represents both tacit and explicit knowledge of the process, also the different types of knowledge conversion. It provides an object library containing the basic objects: information object, task, role, task requirements, person, knowledge object, type of knowledge conversion and knowledge descriptor. The current KMDL 2.2, provides three views: (1) process-based view, (2) activity-based-view (considers the knowledge con-

Table 1: Verification of BPM4KI concepts representation by BPM formalisms.

BPM BPM4KI Concepts		UML AD	eEPC	BPMN 2.0	PROMOTE	Oliveira	KMDL 2.2
Functional Perspective	Collective Action	-/+	-/+	-/+	-/+	-/+	-/+
	Organizational Unit Action	+	-/+	+	-/+	-/+	-/+
	Critical Organizational Activity	-/+	-/+	-/+	-/+	-/+	-/+
	Collaborative Organizational Activity	-/+	-/+	-/+	-/+	-/+	-/+
Organizational Perspective	Agentive Entity	+	-/+	+	-/+	-/+	-/+
	Collective	-/+	-/+	+	-/+	-/+	-/+
	Organizational Unit	-/+	+	+	+	-/+	-/+
	Expert	-	-	-	-	-/+	-
Behavioral Perspective	Control Object	-/+	-/+	+	-/+	-	-/+
Informational Perspective	Material Resource	-/+	-/+	+	-/+	-/+	+
	Input Object	-/+	-/+	+	-	-/+	-/+
	Output Object	-/+	+	+	-/+	-/+	+
	Contingency	-	-	-/+	-	-	-
	Collaboration Protocol	-/+	-/+	+	-	-	-/+
Intentional Perspective	Intention/Distal Intention	-	-/+	-	-	-	-
	Objective	-	-	-	-/+	-	-/+
	Sensitive Business Process	-	-/+	-/+	-/+	-/+	-/+
	External Process	-/+	-	-/+	-	-	-
	Culminated Process	-/+	-/+	-/+	-	-/+	-/+
Knowledge Perspective	Tacit Knowledge	-	-/+	-	-/+	-	+
	Individual Tacit Knowledge	-	-	-	-	-	-
	Collective Tacit Knowledge	-	-	-	-	-	-
	Explicit Knowledge	-	-/+	-	-/+	-/+	-/+
	Individual Explicit Knowledge	-	-	-	-	-/+	-/+
	Collective Explicit Knowledge	-	-/+	-	-	-/+	-/+
	Physical Knowledge Support	-	-/+	-/+	-/+	+	+

versions during the fulfilling of a special tasks), and (3) communication-based view.

Oliveira’s Methodology: The Oliveira’s methodology (Oliveira, 2009) is an extension of Ericsson et al., (2000) for BPM that is composed of diagrams representing a hierarchy of models. It uses constructs adapted from KMDL to model BPs, considering KM aspects.

The evaluation of the BPM formalisms was performed by three experts, who were responsible for observing how well the BPM4KI concepts could be represented in each language. The experts

individually evaluated the correlation between formalism elements and meta-model concepts, considering its definitions and relationships.

3.2 The Evaluation

An overview of the evaluation results can be found in Table 1. Our evaluation scale ranges from comprehensively fulfilled (depicted by +), partially fulfilled (+/-) to not fulfilled (-). The evaluation is carried out from each of the six perspectives making up the BPM4KI meta-model. Results show that none

of the studied formalisms, individually, satisfies SBP modeling requirements.

Generally, the functional and the behavioral perspectives are very well represented in all BPM formalisms, while the organizational and informational perspectives are only partly supported. But a lack of the models is that the knowledge and intentional perspectives are not explicitly supported. In fact, *Expert*, *Individual Tacit Knowledge*, *Collective Tacit Knowledge*, are not addressed at all, in any of the formalisms.

From the process perspective, we can conclude that the traditional BPM formalisms BPMN 2.0 and ARIS eEPC are more expressive for modeling this perspective as a whole. While BPMN offers extended notation for control flow organization, encompasses a high level of detail, numerous constructs (for modeling process logic, decision points, control flows, processes and event types, etc.) offering a very complex expressive model of BPs. In contrast, eEPC has less expressiveness than BPMN, and its constructs are considerable fewer and not so well specified as in BPMN. Furthermore, EPC process models are not intended for being detailed in order to be executed. It is a notation to model the domain aspects of BPs. The focus of the notation is mainly on domain concepts and processes representation rather than the formal specification or technical realization. Besides, the defined concepts-actions specification (Process, Activity, Sub-process, Task, function, action) defined by the selected list of formalisms do not explicitly take into account the individual/collective dimension of the actions. However, taking into consideration such a dimension is very important in our research context, given that we are interested in the localization of knowledge mobilized to realize the BP. This knowledge taken in the action may be either individual or collective/organizational (tacit or explicit).

From the knowledge perspective, knowledge modeling is possible with KMDL 2.2, Oliveira and PROMOTE (including ARIS eEPC, but it incompletely supports this dimension). However, the BP oriented knowledge modeling notation have not been widely adopted by organizations and are very incipient. At the same time they have limited capabilities: (i) They have poor capabilities of process control flow modeling (decisions, actions, control flows, etc.), also they lack the ability to model in an adequate manner the process perspectives as a whole (the structural, behavioral, organizational and informational dimensions); (ii) Information and data concepts are not distinguished; (iii) Most of them do not explicitly differentiate

between tacit and explicit knowledge, which is relevant in SBPs due to, for instance, the high degree of tacit knowledge developed and exchanged among agents through inter-organizational collaboration; (iv) These notations allow modeling knowledge flow perfectly, but BP modeling with KMDL or Oliveira is challenging - understanding a model requires special thinking, learning and significant effort for analysis, design and implementation. They are convenient only for KM experts and require additional training for non-experts.

From the informational perspective, the other notations (BPMN, UML AD and ARIS eEPC) enable data and information modeling, but do not offer a strict border between these terms and are often represented by the same modeling constructs and symbols. It is noteworthy that this distinction is useful and essential for our modeling context. Data and information form the basis for knowledge generation, distribution and utilization in the context of collaboration between BP agents.

To sum up, our evaluation results showed an important loss of information in SBP representations, either because relevant concepts were not addressed by existing formalisms or because these concepts were represented in a very high abstraction level. This may lead to ambiguous and unclear SBP models. The BPMN 2.0 standard was the BPM formalism that presented the broadest coverage of the set of BPM4KI concepts, incorporating requirements for SBP modeling better than other formalisms. Therefore, we select BPMN 2.0 as a basis for representing SBPs to address our research problem, which consists in improving the localization and identification of the crucial knowledge that is mobilized by these processes.

In brief, the best characteristics of BPMN are: (i) BPMN is currently the BP notation most used among strong process modeling practitioners, very simple, easy to use and readily understandable; (ii) BPMN is a BPM standard backed up by OMG, so the language definition is based upon a meta-model built with UML, the notation which is the de facto standard for modeling software engineering artefacts; (iii) BPMN is one of the most recent BPM languages, so it is grounded on the experience of earlier BPM languages, which ontologically makes it one of the most complete BPMLs (Recker et al., 2009); (iv) BPMN is supported by almost all popular BPM tools; (v) BPMN is extensible (with standard extension mechanisms); (vi) BPMN offers a standardized bridge for the gap between the BP design and process implementation, etc.

Nevertheless, despite its strength representation,

some of its concepts should be adapted and extended to be convenient for a rich and expressive representation of SBPs. In fact, this notation does not explicitly support the key concepts of BPM4KI (as *Critical Organizational Activity*, *Individual Tacit Knowledge*, *Collective Tacit Knowledge*, *Expert Knowledge*, *Explicit Knowledge*, *Distal Intention*, *Collective Objective*, etc.). So, to overcoming the shortcomings of BPMN 2.0, this extension must take into consideration, on the one hand, the knowledge dimension, and on the other hand, integrate the new concepts of BPM4KI to represents issues relevant at the intersection of KM and BPM clearly and with a sufficient level of details.

4 CASE STUDY: A SBP MODEL REPRESENTATION

4.1 Case Study Description

In this section, we describe a case study carried out to demonstrate the feasibility, suitability, and practical utility of the evaluated approach to represent and analyze SBP. Precisely, this section illustrates a SBP model using BPMN 2.0, on top of the ARIS express tool (IDS Scheer, 2013) to evaluate its potential in providing an adequate and expressive representation of a SBP, to improve the knowledge localization and identification. The chosen process for this example reflects a medical care process in the Association of Protection of the Motor-disabled of Sfax-Tunisia (ASHMS). This organization is characterized by highly dynamic, unpredictable, complex and highly intensive knowledge actions. Particularly, we are interested in the early care of the disabled children with cerebral palsy (CP) (Ben Hassen et al., 2015b). In fact, the amount of medical knowledge mobilized and produced during this medical care process is very important, heterogeneous and recorded on various scattered sources. One part of this knowledge is embodied in the mind of health professionals. Another part, is preserved in the organizational memory as reports, medical records, data bases, therapeutic protocols and clinical practice guidelines). The created knowledge stems from the interaction of a large number of multidisciplinary healthcare professionals with heterogeneous skills, expertise and specialties (such as neonatology, neuro-pediatrics, physical therapy, orthopedics, psychiatry, physiotherapy, speech therapy, and occupational therapy) and located on geographically remote sites (University hospital of Sfax Hedi

Chaker, University hospital of Sfax, Habib Bourguiba, faculty of medicine of Sfax, research laboratories, etc.). Therefore, the raised problem concerns on the one hand, the insufficiency and the difficulty to localize and understand the medical knowledge that is necessary for decision-making, and on the other hand, the loss of knowledge held by these experts during their scattering or their departure at the end of the treatment. The ASHMS risks losing the acquired know-how for good and transferring this knowledge to new novices if ever no capitalization action is considered. Thus, it should identify the so called «crucial knowledge» to reduce the costs of capitalization operation. Our main objective consists in improving the localization, identification and sharing of different types and modalities of crucial medical knowledge necessary for performing the medical care process of children with CP.

Indeed, this SBP is composed of several sub-processes which consists of a succession of many actions in the form of medical and paramedical examinations and evaluations in different specialties (like neonatology, neuro-pediatrics, physical medicine, orthopedics, psychiatry, physiotherapy and occupational therapy). The different BPs (such as process related to neonatology care, process related to neuro-pediatric care, process related to physiotherapy, etc.) require certain medical information as well as certain medical knowledge (results of para-clinical exams, hospitalization reports, patient-specific knowledge recorded in the medical case file, practice guidelines, etc.).

4.2 The SBP Modeling

In this study, we take into consideration the results of experimentation of the Sensitive Organization's Process Identification Methodology (SOPIM) proposed by Turki et al., (2014a) for the early care of children with CP. As a reminder, the proposed multi-criteria decision making methodology was conducted and validated in the ASHMS organization and aims at evaluating and identifying SBPs for knowledge localization. We have opted for the SBP «Process of neonatology consultation of a child with CP» to illustrate and evaluate the potential of BPMN 2.0 with regard to its applicability and capability of making relevant knowledge embedded in a SBP explicit. Indeed, this SBP is highly dynamic, very complex, in the sense that it involves a large number of organizational units, agents and experts (internal and external who are not affiliated to the organization) from various business/ skills often residing in different physical locations), neonatology

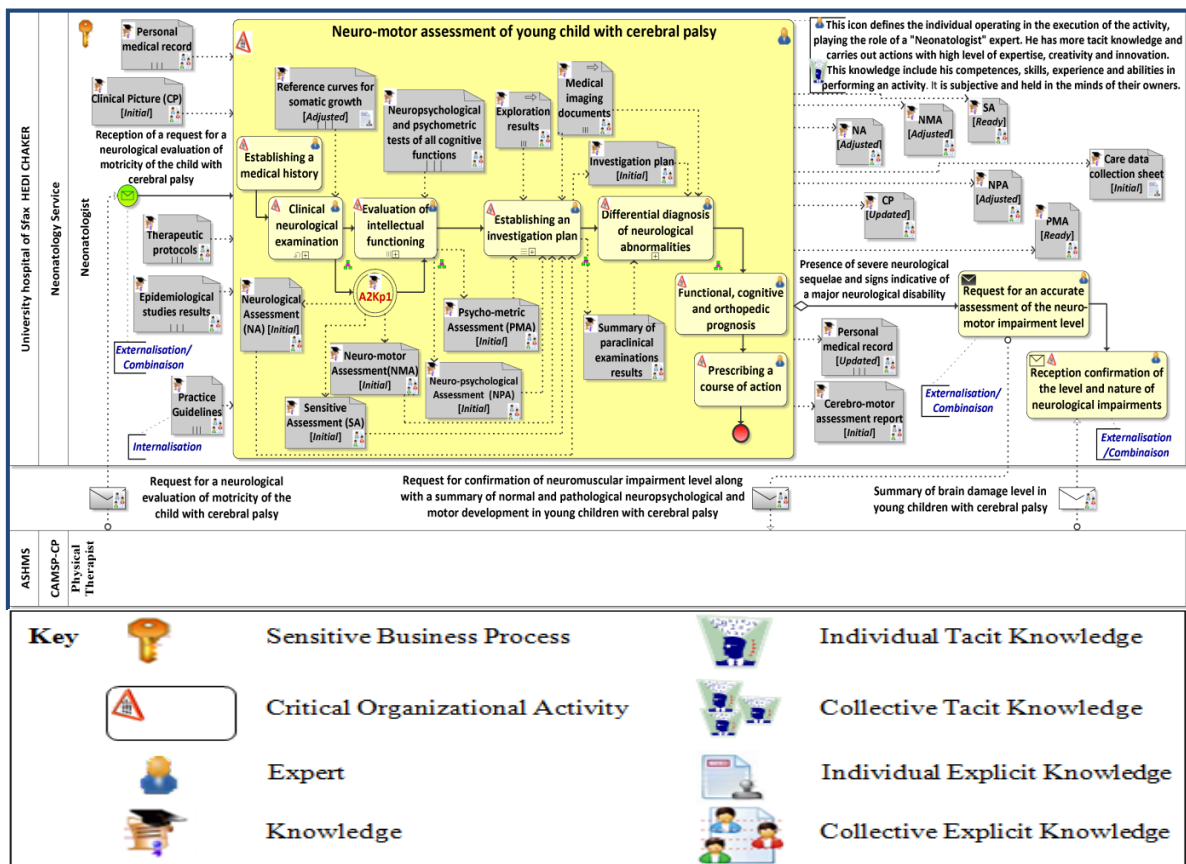


Figure 1: Fragment of SBP model in BPMN related to the neonatology consultation of a child with CP.

disciplines and critical organizational activities (individual and collective). It is very dependent on explicit knowledge sources and on tacit knowledge. In addition, it involves an intense collaboration and interaction between participants to achieve organizational objectives, make decision to deal with an unexpected situation and create value. Some of its activities are highly dependent on the experts experience, expertise and creativity.

Figure 1 outlines a SBP model extract of the neonatology consultation process using BPMN 2.0, enriched with the knowledge dimension (modeled according to BPM4KI). As stated above, this notation does not, however, provide primitives to explicitly represent all relevant aspects related to knowledge dimension in BP models. To remedy for the shortcomings, we tried to extend this notation and started by integrating some specific graphical icons in the form of some BPMN modeling elements relating to several new BPM4KI concepts (Figure 1). The BPMN SBP model is evaluated and validated through some interviews made with 2 stakeholders: the neonatologist and the neuro-pediatrician. During our experimentation, we have

identified different types of medical knowledge mobilized and created by each critical activity related to the SBP of neonatology care. We have distinguished missing or poorly mastered knowledge (individual or collective) necessary to resolve critical problems, expertise, unexplainable tacit knowledge and mastered knowledge necessary and relevant to the proper functioning and development of the activity or produced by the activity. We have also identified the different sources of knowledge, their localization, actors who hold the knowledge, the places where they are usable or used, their nature (like experience, basic knowledge, general knowledge), their degree of formalization (tacit/explicit dimension), their organizational coverage (individual/collective dimension), as well as their quality (perfect or imperfect).

For instance, the knowledge A_2K_{p1} related to « Knowledge about result of the evaluation of the clinical neurological examination, neurological abnormalities, cerebral palsy category, and clinical signs and symptoms associated of young children with cerebral palsy » is produced by the critical activity A_2 « Clinical neurological examination».

Note that this materialized/externalized knowledge is created as a result of the activity execution by the Neonatologist, during which he interacts with information (i.e. source of knowledge information) related to the child with CP (based on his previous experiences and tacit knowledge) to generate and communicate his own knowledge. A_2K_{p1} is stored in the following physical media: the neurological assessment sheet, neuropsychological assessment, the sensitive assessment sheet and the neuro-motor assessment. These physical media of knowledge are located internally within the Neonatology service in the University Hospital Hedi Chaker, precisely in the various archives drawers or patients' directories. A_2K_{p1} is of a scientific, technical and measure nature which is related to patients. It represents a collective explicit knowledge, part of which can be represented in the form of an individual explicit knowledge recorded on the care data collection sheet of the Neonatologist. This knowledge is imperfect (general, incomplete and uncertain). A_2K_{p1} is mobilized by the activity A_3 « Evaluation of intellectual functioning of young child with CP ».

It is important to mention that not all BPM4KI concepts are applicable and must be instantiated in every SBP scenario. Precisely, relevant tacit aspects could not be represented explicitly, such as: the tacit knowledge embedded in the neonatologist's mind, the knowledge conversion and the knowledge flows exchanged between communicating process participants and among activities, and the distal intentions which are responsible for making neonatologist to perform any action and achieve an organizational objective.

Therefore, the relevance of extending BP models with the knowledge dimension (according to BPM4KI meta-model) is manifold:

1. Enhance the localization and identification of the crucial knowledge mobilized and produced by the critical activities: (i) Illustrating the knowledge and its sources that are necessary for the execution of BP activities and are generated, created and/or modified as a results of activities. (ii) Illustrating the knowledge localization (where knowledge can be obtained and clearly stated) as well as experts who hold the (tacit) knowledge. (iii) Illustrating the way in which specific knowledge flows among the activities, or how a specific source is used and modified through the activities. (iv) Illustrating transfers of knowledge between sources, and among activities as well as the different opportunities of knowledge conversion. (v) Defining the knowledge that is being captured or obtained from specific sources. (vi) Giving an opportunity to improve understanding about the knowledge usefulness, validity, and relevance for particular activities in a SBP. (vi) Possibility to evaluate the amount of lost knowledge if a person-owner of knowledge- leaves the organization (i.e., to identify which tacit knowledge in which cases should be transformed into explicit knowledge).
2. A deeply characterizing of the identified knowledge to determine which ones are more crucial to be exploited: (i) Illustrating the nature and degree of formalization of knowledge. (ii) Illustrating the organizational coverage of knowledge, their quality.

5 CONCLUSION AND PERSPECTIVES

This paper presented an evaluation framework of BPM formalisms to represent the SBP, taking the conceptualization defined by BPM4KI (Ben Hassen et al., 2015a) as a baseline. Several BPM notations are reviewed, some are process oriented and some are knowledge oriented. This evaluation concluded that current BPM formalisms are not adequate for the representation of SBPs, since important SBP characteristics details could not be observed. All formalisms were very similar in the number of concepts represented, whereas ARIS eEPC and BPMN 2.0 address the highest representation coverage. In order to observe the practical applicability of the selected formalism, a SBP model of a real neonatology care process of a child with CP was illustrated using BPMN 2.0.

There are several open issues in this paper that we plan to address in the future to deepen the so-called problematic of knowledge identification mobilized by SBPs. Further work is underway to present an extended version of BPM4KI, improving the definition of some BPM4KI concepts so as to ease their understanding, as well as adding new elements to take into consideration issues relevant at the intersection of KM and BPM in greater detail. So, we consider relying on core ontologies (such as core ontology of know-how and knowing-that (COOK) (Ghrab et al., 2015) and Knowledge-Intensive Process Core Ontology (KIPCO) (França et al., 2012). Work for the medium term is to extend the proposed evaluation framework and perform further comprehensive evaluation considering several relevant criteria. Besides the six BP aspects, there are further non-functional requirements a BP meta-model should fulfill: ease of use and

understandability, ontological completeness /expressibility, extendibility/ flexibility, notations, modularity, level of details/granularity, exception handling, pattern support, tools availability, security and privacy, etc. Moreover, the evaluation of six BPM formalisms provides a good starting point that can be easily extended with both further BPM languages and supporting tools. Also, the evaluation will be carried out from each of the six perspectives making up the BPM4KI meta-model. Further down the track, we plan to propose a rigorous scientific approach for extending BPMN 2.0 for KM.

REFERENCES

- Abecker, A., (2008). DECOR Consortium: DECOR-Delivery of Context-Sensitive Organizational Knowledge. E-Work and E-Commerce. IOS Press, Amsterdam.
- Arbeitsbericht, (2009) KMDL® v2.2 (2009). Available at <http://www.kmdl.de/>
- ARIS Expert Paper. (2007) Business Process Design as the Basis for Compliance Management, Enterprise Architecture and Business Rules.
- Ben Hassen, M., Turki, M., Gargouri, F., (2015a). Sensitive Business Process Modeling for Knowledge Management. In *International Conference on Database and Expert Systems Applications (DEXA'2015)*. Valencia-Spain. 2015.
- Ben Hassen, M., Turki, M., Gargouri, F., (2015b). A Business Process Meta-model for Knowledge Identification based on a Core Ontology. *International Symposium on Business Modeling and Software Design (BMSD'2015)*. Milan, Italy. Selected best Papers. Lecture Notes in Business Information Processing, Springer.
- Cabral, L., Norton, B., Domingue, J., (2009). The business process modelling ontology. In *4th International Workshop on Semantic Business Process Management (SBPM 2009)*. Workshop at ESWC' 09, Crete, June.
- Donadel, A.C., (2007). A method for representing knowledge-intensive processes. *M.Sc. dissertation. Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento*, Universidade Federal de Santa Catarina, Brazil.
- França, J.B.S., Netto, J.M., Barradas, R.G., Baião, F. A., Santoro, F. M., (2012). Towards Knowledge-Intensive Processes Representation. In *International Workshop on Business Process Design*. Estonia.
- Ghrab, S., Saad, I., Kassel, G., Gargouri, F., (2015). An ontological framework for improving the model of contribution degree of knowledge. In *International Conference on Knowledge Management, Information and Knowledge Systems (KMIKS 2015)*, Tunisia, pp. 45–58.
- Gronau, N., Korf R., Müller C., (2005). KMDL-Capturing, Analyzing and Improving Knowledge-Intensive Business Processes. *J. of Universal Computer Science*, Vol. 11, 452–472.
- Grundstein, M., (2000). From capitalizing on Company Knowledge to Knowledge Management. In *Knowledge Management, Classic and Contemporary Works by Daryl Morey, Mark Maybury, Bhavani Thuraisingham*, Cambridge, Massachusetts. The MIT Press, Chapter 12, pp. 261-287.
- Heisig, P., (2006). The GPO-WM® method for the integration of knowledge management into business processes. In *International Conference on Knowledge Management*, Graz, Austria, pp. 331–337.
- Hildebrandt, T.T., Mukkamala, R.R., (2010). Declarative event-based workflow as distributed dynamic condition response graphs. In *Programming Languages Approaches to Concurrency and Communication-centric Software*, p.59-73, Cyprus.
- Korherr, B., List, B., (2006). A UML 2 profile for event driven process chains. In: *Proceedings of the 1st IFIP International Conference on Research and Practical Issues of Enterprise Information Systems*. Springer.
- Lin, FR., Yang, MC. and Pai, YH., (2002). A generic structure for business process modeling. *Business Process Management Journal*, Vol. 8. No.1, Emerald.
- List, B., Korherr, B., (2006). An evaluation of conceptual business process modelling languages. In *ACM Symposium on Applied Computing (SAC' 06)*. ACM Press, France
- Netto, J.M, Franca, J. B. S., Baião, F.A., Santoro, F. M., (2013). A notation for Knowledge-Intensive Processes. In *IEEE 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD 2013)*, Vol.1, pp.1–6.
- Oliveira, F.F., (2009). Ontology Collaboration and its Applications. MSc Dissertation. Programa de Pós-Graduação em Informática, Universidade Federal do Espírito Santo, Vitória, Brazil.
- OMG (2011a). Business Process Modeling and Notation (BPMN). Version 2.0. <http://www.bpmn.org/>
- OMG (2011b). Unified modeling language (UML). Version 2.0. <http://www.uml.org/>
- Papavassiliou, G., Mentzas, G., (2003). Knowledge modelling in weakly-structured business processes. *J. Know. Manag.* 7(2), 18–33.
- Prezel, V., Gavsević, D., and Milanović, M., (2010). “Representational Analysis of Business Process and Business Rule Languages. In *Proceedings of the 11th IEEE International Conference on Enterprise Distributed Object Computing Conference*, Annapolis, Maryland, USA, pp. 241–258.
- Schreiber, G., Akkermans, H., Anjewierden, A., Hoog, R., Shadbolt, N., De Velde, W.V., Wielinga, B., (2002). *Knowledge Engineering and Management: The Common KADS Methodology*. MIT Press, Cambridge
- Strohmaier, M., Yu, E., Horkoff, J., Aranda, J., Easterbook, S., (2007). Analyzing knowledge transfer effectiveness an agent-oriented modeling approach. In *Proceedings of the 40th Hawaii Int. Conference on System Sciences*, USA.

- Söderström, E., Andersson, B., Johannesson, P., Perjons, E., Wangler, B., (2002). Towards a Framework for Comparing Process Modeling Languages, *Advanced Information Systems Engineering. 14th International Conference, CAiSE 2002*. Toronto, Canada.
- Sultanow, E., Zhou, X., Gronau, N., Cox, S., (2012). Modeling of Processes, Systems and Knowledge: a Multi-Dimensional Comparison of 13 Chosen Methods. *International Review on Computers and Software*, 7(6), pp. 3309-3319.
- Recker, J., Rosemann, M., Indulska, M., Green, P., (2009). Business process modeling: a comparative analysis. *Journal of the Association for Information*, V (10), 333-363.
- Saad, I., Grundstein, M., Sabroux, C., (2009). Une méthode d'aide à l'identification des connaissances cruciales pour l'entreprise. *Revue SIM*, Vol 14, n° 3.
- Scheer, A.W., (2000). *ARIS – Business Process Modeling*. Springer.
- Schlenoff, C., Gruninger, M., Tissot, F., Valois, J., (2000). The process specification language overview and version 1.0 specification. <http://www.mel.nist.gov/psl/>
- Supulniece, I., Businska, L., Kirikova, M., (2010). Towards extending BPMN with the knowledge dimension. In *BPMDS 2010 and EMMSAD 2010*. LNBI, vol. 50, pp. 69–81. Springer, Heidelberg.
- The IDS-Scheer website (2013). <http://www.ids-scheer.com/>
- Turki, M., Saad, I., Gargouri, F., Kassel, G., (2014a). A Business Process Evaluation Methodology for Knowledge Management based on multi-criteria decision making approach. *Information systems for knowledge management*. ISBN:978-1-84821-664-8, Wiley-ISTE.
- Turki, M., Kassel, G., Saad, I., Gargouri, F., (2014b). COOP: A Core Ontology of Organization's Processes for group decision making". *Journal of Decision Systems*, 23(1) pp.55-68.
- Wand, Y., Weber R., (1990). An ontological model of an information system. *IEEE Transactions on Software Engineering*, 16(11):1282–1292.
- Weidong, Z., Weihui, D., (2008). Integrated Modeling of Business Processes and Knowledge Flow Based on RAD. In *International Symposium on Knowledge Acquisition and Modeling*. Wuhan, China, p. 49-53.
- Woitsch, R., Karagiannis, D., (2005). Process Oriented Knowledge Management: A Service Based Approach. *Journal of universal computer science* 11(4), 565-588.

A Proposal to Model Knowledge in Knowledge-Intensive Business Processes

Norbert Gronau, Christoph Thim, André Ullrich, Gergana Vladova and Edzard Weber
Business Informatics, esp. Processes and Systems, Potsdam University, August Bebel Str 89, Potsdam, Germany
{ngronau, cthim, aullrich, gvladowa, eweber}@lswi.de

Keywords: Knowledge, Business Process, Ability to Articulate, Generality, Information.

Abstract: This paper addresses the topic of modeling tacit knowledge across business processes. Some approaches exist to cover that issue but none is really satisfying. Therefore a new approach is proposed, which is based on more than ten years of experience with the Knowledge Modeling and Description Language (KMDL). The new approach suggests to differentiate knowledge in professional insight, experience and context and to describe the degree of ability to articulate and generality.

1 INTRODUCTION

This paper addresses the topic of modeling tacit knowledge across business processes. Some approaches exist to cover that issue but none is really satisfying. Therefore a new approach is proposed, which is based on more than ten years of experience and also overcomes the deficits of existing approaches.

One of the often used definitions of tacit knowledge is based on Davenport's set of criteria that consists of information, professional insight, values, experience and context (Davenport and Prusak, 1998; Gronau, 2012). Conventional approaches for the differentiation of knowledge types like Polanyi (1966) or for the differentiation of the handling of knowledge like the SECI model (Nonaka and Takeuchi, 1995) see tacit knowledge only bound to humans. That might be too narrow in the light of new cyber-physical systems as self-organizing and decision-capable technical entities (Lee et.al, 2014; Gronau, 2015). In the future at least some of the competencies to make decisions will lie with technical actors.

Digitalization, virtualization and the Internet-of-things force great changes in the roles of the employees and the technical actors. Machines and factory units collect data from their environment with the help of sensors, process these data and act in the environment using mechanical actuators. Data will be sent to information systems, which receive, process and forward them. This is an analogy to the human

information processing. Processing includes the use of information following predefined rules and a predefined space of alternative solution paths (Inference), and the creative development of facts and solutions additionally to predefined structures with a not predetermined result (intelligence, cf. Turing, 1950).

Knowledge as a „goal-oriented netting of information“ (Rehauser and Kremer, 1996) allows that actors to act and to decide. It helps to prepare decisions and is an important component to generate competencies. The netted information contains data with semantics and data with a certain syntax. Human as well as technical actors are able to proceed signs, data information and knowledge with existing technology. Therefore it might be useful to see also the technical entities as potential bearers of knowledge. While value creating processes become more and more interwoven with cyber-physical systems, some of the concepts developed for person-bound knowledge also can be used for a machine's knowledge. Especially the aspects of professional insight in a specific domain and the experience are candidates for a transfer from man to machine. Experience for instance can a machine gain and process by using a case-based-reasoning system.

Another problem occurs when the usage of knowledge in teams is investigated. This kind of knowledge cannot be characterized with the criteria given by Davenport, because most of these criteria are only suitable for one human.

What makes the difference between subjective knowledge and explicable and more objective

knowledge? It is the context, in that the information is announced and it is the value that may be very different for two humans looking at exactly the same piece of information. Therefore the authors of this contribution see information as an extreme occurrence of knowledge. During the process of explication the context was reduced and the value propositions were omitted.

One of the modeling techniques that is able to represent knowledge bound to persons is the Knowledge Modeling and Description Language (KMDL®). Its development started more than ten years from now. In that time a lot of experience was gained, especially in the areas of software development, product development, innovation processes, quality management and other areas (Gronau, 2012). Based on these experiences the authors suggest to differentiate knowledge following the criteria of professional insight, experience and context and to look at generality and ability to articulate for each of these criteria. The following sections describe this proposal in more detail.

2 THE TERM KNOWLEDGE

Stemming from the complexity of the term knowledge the necessity occurs to differentiate in knowledge types and knowledge dimension. The supposedly most often used differentiation discriminates between tacit and explicit knowledge. The tacit dimension was first described by Polanyi and addresses parts of the personal knowledge, which are neither to be scribed nor to be articulated. *„Although the experts (···) can indicate their clues and formulate their maxims, they know many more things than they can tell, knowing them only in practice, as instrumental particulars, and not explicitly, as objects.“* (Polanyi 1958, S. 88)

Tacit knowledge is „personal, context specific and very difficult to communicate“ (Nonaka and Takeuchi 1995, p. 72). Contrarily explicit knowledge can be distributed in a formal and systematic language. Tacit knowledge can be seen as a synonym of embodied and procedural knowledge (Meyer and Sugiyama, 2007, p. 26).

Davenport and Prusak (1998, p. 5) deliver a so-called pragmatic definition of knowledge: *„Knowledge is a fluid mix of framed experience, values, contextual information,*

and expert insight that provides a framework for evaluating and incorporating new experiences and information. It originates and is applied in the minds of knowers.“

Knowledge is seen as very difficult to articulate and also person-bound. It is based on information but cannot be equaled with it. To make the term knowledge more comprehensible, Davenport and Prusak (1998) refer to six key components: experiences, ground truth, complexity, judgment, rules of thumb and intuition, values and beliefs.

Explicit and tacit (some authors use the wrong term of „implicit“) knowledge are defined by pointing out the difference in processing these two knowledge types. Explicit knowledge can be transferred by communication, by numbers, pictures or language. It can be processed, altered and learned together (Willke 2001; Franken and Franken 2011, p. 33).

Lam (2000) has given a description of knowledge that refers not only to qualities but also to the organizational context: The encoded knowledge has an existence independent of persons and can be stored in handbooks, data bases, rules of conduct etc. and can be seen as organizational explicit knowledge (see also Blackler, 1995). The embedded knowledge to the contrary cannot be transferred objectively but is socially constructed, captured in organizational cultures, language systems etc and used and shared by the members of the organization. Different types of knowledge are differentiated in the realm of organizational knowledge:

- encultured knowledge, which is shared by the members of the organization and transferred by socialization (Sackmann, 1991; Kogut and Zander, 1992)
- event knowledge that is concerned to events in the lifetime of the organization (Vlaar et al, 2007)
- procedural knowledge about processes and connections (Fischer, 2008).
- embodied knowledge describes the dimension of individual tacit knowledge. It is bound to persons and can only be created by experience (Polanyi, 1966; Blackler, 1995; Nonaka and Takeuchi, 1995).

Franken and Franken (2011, p. 30) say that knowledge is something immaterial, difficult to describe, but with great influence on human acting. It has to be distinguished between the real world on the one hand and the immaterial world of knowledge on the other hand, which exist in the human brain as a result of experiences and learning, leading to mental patterns. In this way knowledge is developed as an

individual construction from the interaction with the real world (Franken and Franken, 2011, p. 31). Rehauser and Krcmar (1996) denote knowledge as an individually modeled reality, which is generated by the bearer of knowledge under the influence of her own perspective. Knowledge allows to act and the artifacts created during the action cause a change in the real world, induced by the individual person.

Summarizing it can be reasoned that the term “tacit knowledge” encompasses a broad area of different characteristics. This makes its transfer into a model, which is necessary to get a grip on knowledge processes and knowledge conversions, very difficult. In the following sections the deduction of such a model in the context of KMDL is presented.

3 DEDUCTION OF A CONCEPT FOR THE MODELING OF KNOWLEDGE

Starting with the different characteristics of the term tacit knowledge a classification is necessary as a first step. First knowledge can be classified following the definition of Davenport and Pruzak (1998, p. 5). Following them knowledge consists of experience, values, context information and professional insight.

Experiences stem from a practical engagement with a certain topic. Professional insight is the intellectual penetration of an area of content. Values are generated by socialization procedures and are shared conjointly. They are deeply embedded into one’s personality. On the other side context information is the picture of an observation. This observation can relate to an object, a person, a topic of the environment or a self-observation.

Existing approaches that recognize these differentiations and the relations between the components are rare (Hinkelmann et al, 2002; Heisig, 2000; Allweyer, 1998; Gronau and Froeming, 2006). Following the guidelines of proper modeling (Becker et al, 1998) the possibility and usefulness of every component has to be judged. The six requirements are relevancy, correctness, economic feasibility, clearness, comparability and systematic construction.

The question of relevancy was solved by selecting criteria and justify their selection above. The other requirements are combined to judgment criteria.

The requirements correctness and economic feasibility are merged into the criterion ascertainability. Ascertainability states whether components of knowledge can be grasped objectively true and whether this is possible with reasonable

effort. A modeling of a component of knowledge is only possible if this component can be captured by an observer or by self-observation.

The requirements clearness and comparability are combined into the criterion intersubjective comparability. The intersubjective comparability also is important to be able to compare certain model statements and to be able to model the transfer of knowledge.

These components of knowledge can be captured and compared in different degrees of easiness, as seen in Table 1.

Table 1: Judgment of easiness of modeling.

Component	Ascertainability	Intersubjective comparability
Professional insight	++	++
Experience	++	++
Values	o	--
Context	+	+
++ very good, + good, o no statement possible, - bad, --very bad		

Professional insight, for instance in the shape of formal education, can be captured by certificates or the documentation of training periods. These are also comparable very good, by certificate degrees, age of knowledge etc.

Experience can be captured objectively by documenting core areas of action or by self-judgment. Although distortions are possible, typically the results are mostly correct. Also an intersubjective comparability is given, when durations, frequency or intensity of actions are compared between different knowledge bearers.

Values are very difficult to capture due to their often un-reflected anchoring in the human consciousness and their very subjective character. Also an intersubjective comparison between values is not possible, because it is very difficult to create a hierarchy of values or to compare the value systems of two humans. A pure description of equal or different values is not suitable for the modeling purpose. Another argument is that the dissemination of values in an organization occurs over time and is of long duration, therefore not usable in the context of process-oriented knowledge management. Properties of values that are relevant for decisions can be modeled in the context component.

The context component can be captured in a sufficient manner when the usage environment is

described or observed. Although the context can be compared inter-subjectively, different interpretations or perspectives can occur.

Following those thoughts, values cannot be modeled sufficiently. The remaining components to model knowledge objects are therefore professional insight, experience and context.

Knowledge seldom can be assigned only to one single component. The judgment of the context typically uses experience. Capturing of professional insight is done within a context and the collection of experience only works when professional insight is available. Therefore these components have to be inspected together depicting the knowledge of person, an item or a status.

Beside the differentiation of knowledge components to be able to model the use and the transfer of knowledge more information is necessary. For a more detailed description the knowledge dimensions of Spinner (2002) can be used. He differentiates the shape, expression, content and validity dimensions.

Table 2: Judgment of modeling of knowledge dimensions.

Component	Ascertainability	Intersubjective comparability
Shape	++	++
Expression	+	+
Content	+	--
Validity	-	+

++ very good, + good, o no statement possible, - bad, --very bad

The shape dimension indicates the generality of knowledge on a scale between particular and general. The expression dimension depicts the degree of articulation and has the polar characteristics tacit and explicit. The content dimension indicates how much information lies in the knowledge, between nearly and highly informative. The validity dimension shows how much the knowledge is backed by facts or scientific results. This dimension has the polar characteristics of hypothetical and apodictic.

Again the dimensions can be checked with their degree of ascertainability and intersubjective comparability to judge the transfer into knowledge modeling (Table 2).

Following Table 2 we can see that especially the dimensions of shape and expression are suitable to integrate into modeling. The content dimension cannot be compared inter-subjectively, due to different prevalent knowledge and different interest in the subject. The content dimension is therefore

different between two persons and during different points in time. Additionally no judgment of the value propositions of the bearer of knowledge is intended, especially because it is very difficult to measure a value proposition. Nevertheless the authors want to state that the attached value remains an important part of the description of person-bound knowledge.

Further on the validity dimension is difficult to capture on an individual level. Whether some element of knowledge is hypothetical or rock solid cannot be determined in most cases.

The concentration of the two remaining dimensions allow a more detailed description of knowledge. Both dimensions can be applied on the components so that a 2x3 matrix is constructed (Table 3).

Table 3: Characteristics of a knowledge object.

	Professional insight	Experience	Context
Ascertainability	[0,1]	[0,1]	[0,1]
Generality	[0,1]	[0,1]	[0,1]

Professional insight, experience and context are judged referring to generality and ascertainability by the bearer of the knowledge with values from 0 to 1.0 means, there is no expression of this characteristic while 1 means there is a maximal expression of this characteristic. For the dimensions it means as follows:

Ascertainability:

- 0 - not articulable, real tacit knowledge
- 1 - completely articulable

Generality:

- 0 - particular, only useful in a single instance
- 1 - commonly useful

Instead of the suggested numbered scales also other scales are possible, so for instance pure yes-no-depictions or judgments like low - medium - high.

Using these six characteristics, a very detailed differentiation of a knowledge object can be processed. Therefore it is suggested to use this new knowledge object while modeling with KMDL (Figure 1).

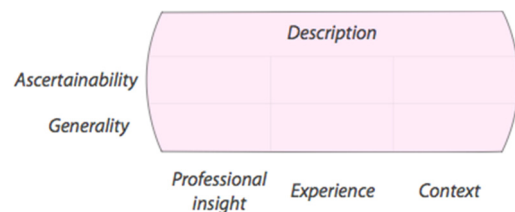


Figure 1: Knowledge object in KMDL.

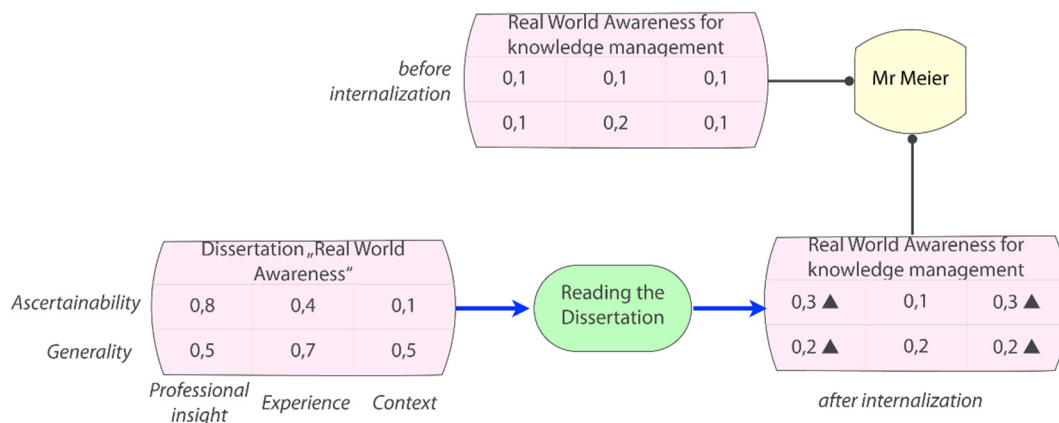


Figure 2: Modeling of internalization of knowledge.

After introducing the multi-dimensional of knowledge objects a differentiation between knowledge and information objects can be omitted. Therefore in the activity view of MDL only knowledge objects are shown and the information objects move to the process view to assure comparability to other BPM modeling approaches.

An additional advantage lies in the better ability to interpret the conversion of knowledge. This is explained using two examples:

Example 1: ERP usage in chemical industry

An expert of ERP systems in the chemical industry can articulate her knowledge with a degree of ascertainability of 0.85. Under some circumstances she will get her knowledge about that topic from books and journal essays, but not only from practical experience. Therefore she has a great ability to articulate but a quite limited experience.

Example 2: Vegan food

Now the same expert from example 1 shall speak about vegan food. Due to missing personal experiences but because the ascertainability of the expert her knowledge can be assessed, Low values for generality mean that her knowledge is not very useful for others, although she is able to articulate it quite good.

Beside the better representation of the knowledge of certain actors in the process also the knowledge conversions externalization, internalization socialization and combination ca be represented better. The modeler has to decide about his point of observation and about the purpose of the modeling beforehand. By comparing the scale expressions of the bearer of the knowledge object before and after the conversion also an increase of knowledge can be

measured - clearly a real advantage against other modeling approaches!

In Figure 2 an internalization is depicted using the newly developed knowledge object. Not the transfer of knowledge from the printed dissertation to the knowledge of he bearer is of interest here but the increase of the bearer’s knowledge about the topic before and after reading the dissertation. This can be seen that in four of the six characteristics of a knowledge object an increase took place and only two characteristics remain unchanged.

4 CONCLUSIONS

Modeling the occurrence of knowledge is the decisive key to be able to recognize potentials for the improvement of knowledge-intensive business processes. For this purpose a differentiation of knowledge is very important. This paper proposed a framework to capture only these dimensions of knowledge which can truly being captured during modeling.

ACKNOWLEDGEMENTS

The authors would like to thank Marcus Grum for his many valuable suggestions.

REFERENCES

Allweyer T, 1998. Knowledge Management with ARIS Models. In: Scheer A-W: ARIS - From Business process to the application system (in German), Heidelberg

- Becker J, Rosemann M; Schuette R, 1995. Guidelines of proper modeling (in German). In: *Wirtschaftsinformatik* (37) 5, 435-445.
- Blackler F, 1995. Knowledge, Knowledge Work and Organizations: An Overview and interpretation; in: *Organization Studies*, 16(6),1021-1046.
- Davenport T, Prusak L, 1998. *Working knowledge: How organizations manage what they know*, Harvard Business School Press, Boston.
- Fischer M, 2008: Organizational learning in Corporations with extreme performance pressure (in German). In: Pawlowsky P; Mistele P (Eds.): *High performance management* (in German), Wiesbaden: Gabler,111-131
- Franken R, Franken S, 2011. *Integrated knowledge and innovation management* (in German). Wiesbaden: Gabler.
- Gronau N, 2015. The influence of Cyber-Physical Systems on the creation of production systems (in German). *Industrie Management* 3, 16-20
- Gronau N, Heinze P, Weber E, 2012: Conversion- and Transformation-Oriented View on Knowledge Flows. In: Gronau N (ed): *Modeling and Analyzing knowledge intensive business processes with KMDL - Comprehensive insights into theory and practice*, Berlin, 249–268.
- Gronau N, Fröming J, 2006: A semi-formal modeling language to describe knowledge conversions (in German). In: *Wirtschaftsinformatik* (48) 349-360.
- Heisig,P, 2002. GPO-WM - Method and tool for process oriented knowledge management (in German). In: Abecker A, Hinkelmann K, Maus H, Mueller HJ(eds.): *Process-oriented Knowledge management* (in German), Berlin Heidelberg, New York, 47-64.
- Hinkelmann K, Karagiannis D, Telesko R, 2002. PRO-MOTE - Method and tool for process oriented knowledge management (in German). Abecker A, Hinkelmann K, Maus H, Mueller HJ (eds.): *Process-oriented Knowledge management* (in German), Berlin Heidelberg, New York,
- Kogut B, Zander U, 1992. Knowledge of the Firm: Combinative Capabilities and the Replication of Technology; in: *Organization Science*, Vol. 3, No. 5 (August), 383-397
- Lam A, 2000. Tacit Knowledge, Organizational Learning, and Societal Institutions: An Integrated Framework. *Organization Studies*, vol. 21 (3), 487-513.
- Lee J, Bagheri B, Kao H-A, 2014. A Cyber Physical Systems Architecture for Industry 4.0-based Manufacturing Systems. *Manuf Lett* 3, 18–23
- Meyer B, Sugiyama K, 2007. The concept of knowledge in KM: a dimensional model", *Journal of Knowledge Management* (11) 1, 17 - 35
- Nonaka I, Takeuchi H, 1995. *The knowledge creating company: how Japanese companies create the dynamics of innovation*. New York: Oxford University Press, 284
- Polanyi M, 1966. *The tacit dimension*. Garden City, Doubleday
- Polanyi, M, 1958. *Personal Knowledge: Towards a Post-Critical Philosophy*. University of Chicago Press, Chicago, IL.
- Rehauer J, Krcmar H, 1996. Knowledge Management in the corporation. In: Schreyoegg G, Conrad P (eds.) *Management research 6: Knowledge Management*. Berlin, 1-40
- Sackmann S, 1991: *Uncovering Culture in Organizations*; in: *Journal of Applied Behavioral Science*, 27, 295-317.
- Spinner HF, 1994. *The knowledge order. A guiding concept for the third order of the information age* (in German) Opladen
- Turing AM, 1950. *Computing machinery and intelligence*. *Mind*, 433-460.
- Vlaar P, Van Den Bosch F, Volberda H, 2007. Towards a Dialectic Perspective on Formalization in Interorganizational Relationships: How Alliance Managers Capitalize on the Duality Inherent in Contracts, Rules and Procedures; in: *Organization Studies*, Vol. 28, 437 - 466
- Willke H. 2001. *Systemic Knowledge Management* (in German). Stuttgart. 2. edition.

Characteristics of Enterprise Architecture Analyses

Julia Rauscher, Melanie Langermeier and Bernhard Bauer

Software Methodologies for Distributed Systems, University Augsburg, Augsburg, Germany
{rauscher, langermeier, bauer}@ds-lab.org

Keywords: Enterprise Architecture, Analysis, EA Analysis, Categorization, DSL.

Abstract: Enterprise Architecture Management (EAM) deals with the assessment and development of business processes and IT landscape of an organization. Analyses are an important mean in the EAM process. They support the understanding of the architecture and evaluate the current situation as well as possible future ones. In current literature exists various different approaches to EA analyses. Each pursues different goals and utilizes different techniques. We evaluated current literature about EA analyses and identified possible categories. Therefore we define requirements for an EA analysis and utilize them for a classification of the various approaches. We propose a two-dimensional classification approach. Technical categories cluster analyses according their procedure and utilized techniques. Functional categories cluster analyses according to their goals and outcome. To validate our categorization and the analysis requirements we develop a domain specific language, which is used to formalize the existing analysis approaches from literature.

1 INTRODUCTION

Analyses are one of the most important artifacts integrated in Enterprise Architecture Management (EAM) and are indispensable in the EAM cycle. The EA process contains five phases (Niemann, 2006): Document, Analyze, Plan, Act and Check. Thus, analysis is an essential part in order to create and implement future plans. It supports decision making through an evaluation of the current architecture as well as potential future scenarios (Sasa and Krisper, 2011). The result of analysis and planning actions is finally the creation of a target architecture. Those actions enable also planning, acting, controlling and documenting through all layers.

The creation of an Enterprise Architecture (EA) model is time and cost consuming. Therefore, support for decision making and planning generates value and increases the acceptance of the EA initiative in an organization (Lankhorst, 2013). Thus, analysis support is essential in order to generate value from an EA model. The execution of an analysis decomposes the analyzed object in its components. Those single elements are examined and evaluated as well as the relationships and interactions between them. Applying existing analysis on established models is expensive, since the corresponding meta models typically require some adaptations (Langermeier et al., 2014). This makes reuse of existing solutions and research findings hard.

In current literature a great variety of different analysis possibilities are described. They rely on different technologies, like probability networks (Närman et al., 2008), ontologies (Sunkle et al., 2013) or expert interviews (Kazienko et al., 2011), have different preconditions and provide different kind of results. E.g. preconditions can be required properties for model elements or specific data structures. Typical results are quantitative ones like an overall metric for the architecture, measures for specific architecture elements, but also a determined set of elements. Every analysis supports a different goal and thus, for a sound evaluation of the architecture different kinds of analyses are required.

In this paper we want to provide a categorization of existing EA analyses from literature. The main goal of the categorization is to create a possibility to conduct analyses organized and controlled, to create new analyses and to choose the best suited analysis depending on the goal and requested technique. We study existing analyses regarding their requirements for execution and their provided result. This provides us a sound overview of analysis approaches in the context of EA and of the issues they address. Based on the identified similarities we establish a categorization of the approaches. Once according to their functional dimension, and once according to their technical dimension. The resulting categories with their characteristics are evaluated through the establishment of a Domain Specific Language (DSL).

This allows us to formalize them and validate their correctness by modeling existing analyses. Additionally such a language allows us to make the requirements of an analysis visible in a structural way. The calculated outcome as well as the preconditions in order to execute the analysis are easily accessible.

The remaining paper is structured as followed. First we introduce foundations of EA analysis (section 2). Following we present in section 3 our approach for determining the analysis categories. The categories itself are also presented shortly with their main characteristics. The DSL to describe the analyses is introduced in section 4.1. Its application is shown exemplary for one category. Finally we discuss the categorization and the DSL (section 4.2).

2 ENTERPRISE ARCHITECTURE ANALYSIS

Architectures are used to describe the elements of a system and the relationships between them. The term also comprises the process of creation and maintenance, the architecture work (Lankhorst, 2013). Often used layers are the strategic layer to represent the organization's strategy with its goals, the business layer describing the business processes and products, the information layer with the information objects, and the application layer as well as the infrastructure layer describing the Soft- and Hardware components (Lankhorst, 2013; Winter and Fischer, 2006). Despite the examination of different layers the focus of an EA are the dependencies between layers, i.e. how business and IT relate to each other. Layers are dependent according to the Align-Enable-Principle. The lower layers are the foundation for the upper ones, and the upper ones adjust the lower ones (Winter and Fischer, 2006; Krcmar, 2015).

The main reasons of analyzing EA is to receive an overview of the whole architecture, its components and connections, and to analyze the as-is state (Langermeier et al., 2014). Furthermore weak points can be revealed, new advantages be discovered and various design alternatives be evaluated (Zia et al., 2011). The result of analysis activities is the development of a to-be architecture as well as improved decision making. The focus of every analysis depends on the analysis type. Additional the questions of what is feasible and what is desirable are crucial (Johnson et al., 2007). The process of analysis can be segmented in different phases and activities (Wan and Carlsson, 2012). We used the parts "system thinking", "modeling", "measuring", "satisfying", "comparing with requirements" and "comparing alternatives" in

this work to identify characteristics of analysis categories.

To receive a basis for our work we conducted a detailed literature research (Rauscher, 2013; Rauscher, 2015). We exclusively chose analyses, which purely analyze EA and are not transferred from other topics. Hereof a pure EA analysis has the focus on collecting data and discovering the current state of an enterprise architecture in a quantitative or functional way to create a summary, alter the state or control different aspects. The goal of this selection was to create an overview of current EAM analyses and to receive approaches utilizable for a categorization (e.g. Della Bordella et al., 2011; Johnson et al., 2007; Razavi et al., 2011). We identified 105 EAM analyses which are roughly grouped into 40 EA analysis types in previous work (Rauscher, 2013). An analysis type describes analyses which have the same rough scope and are built independent on the realization method. The goals of contained analyses can differ significant. Examples of these types are 'Quality Analysis' (Närman et al., 2008), 'Requirements Analysis' (Aier et al., 2009) and 'Analysis of Costs' (Niemann, 2006). The different types of analyses which have been discovered in the literature research can be treated as a first categorization. However this categorization only makes raw statements about the rough purpose of the contained analyses. Although analysis of the same analysis type have the same field of interest their individual goals and implementations can differ. It's not detailed enough to derive characteristics and the covered analyses don't have to share them. Quality analyses, for example, can be conducted in various ways and can target different goals from quality of a whole system to maturity quality of a single artifact.

3 CATEGORIZATION

The huge amount of different approaches clarifies importance of EA analyses and coherence of a successful architecture. To ease the usage of analysis and get an understanding about current work we categorized them according their characteristics. We define characteristics of an EA analysis as all necessary steps and components of an analysis to accomplish its goal. The characteristics are a main part of the categorization because they are guaranteeing the accuracy of the conducted analysis and the achievement of the target. Figure 1 gives an overview over the categorization approach, which is described in detail in sub-section 3.1. The resulting categories are presented in sub-sections 3.3 and 3.4.

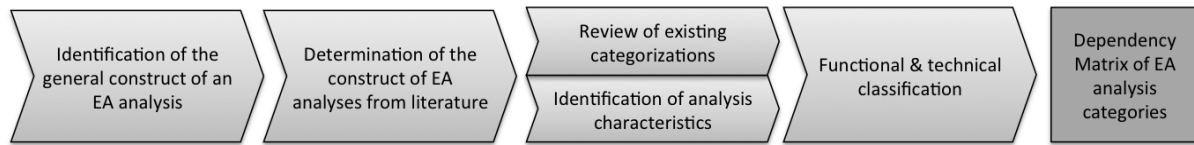


Figure 1: Categorization approach

3.1 Categorization Approach

Preliminary work for the categorization includes the definition of a general understanding of an analysis to determine a general purpose construct. We could identify three main constructs of an EA analysis, which are used as foundation for the categorization and determination of characteristics. These are data *intake*, *processing* and *outcome*. Other parts vary per analysis. Based on these parts we determine the meaning and boundaries of a category: Analyses can be merged to a category if at least one of the three parts given above is similar. In an optimal condition all parts are equal, but this condition is usually not given. Therefore we classify different analyses to the same category if they have at least the same target or same processing technique.

After defining our basis we conduct literature research of current EA analysis (procedure see section 2). We determined the construct for each of the identified EA analysis approaches. Thereby we ensured that only paper with a high elaboration level are used. Because of missing details it's not possible to analyze rough approaches and to identify their construct and characteristics. For elaborated analyses with less detailed parts, we made necessary assumptions. In the case that an EA analysis approach is realized using another non-EA related analysis approach, this non-EA analysis is included too. This proceeding ensures the construction of a data basis with categorize-able analyses according to the general construct of intake, processing and outcome.

Based on the experiences made while identifying the construct of the EA analysis we can refine our categorization approach. Considering different existing kinds of categorization (Lankhorst, 2013; Buckl et al., 2009b) we conducted our approach with two main fields of categories: functional and technical. This decision brought the most advantages in comparison with other approaches because of the division in "How" (technical aspects) and "Why" (functional purpose). The additional distinction in architecture levels is not included in our approach because a plain allocation wouldn't be possible. Most analyses can be conducted in many levels or can only be performed by involving other levels. Through the new and detailed knowledge from the first evaluation of EA analysis

constructs we introduce characteristics to ensure accuracy. As only properties and steps can show the components responsible for classification, we used these characteristic kinds to analyze the approaches again to receive detailed information of their single categories.

After this step the final categorization of the analyses was received based on our main idea of distinction between functional and technical. The business functions of every analysis are determined based on the concepts *purpose dependent division* ("Fundamental", "Main" and "Decision-oriented") and *activity dependent division* ("System thinking", "Modeling", "Measuring", "Satisfying", "Comparing with requirements" and "Comparing alternatives") (Wan and Carlsson, 2012). We used these concepts to analyze the identified analysis approaches according to their goals and activities. Thereby the functional categories have been determined by using a prepared template of aspects. This template consists of the analysis activities, the intermediate objectives and the main goal. After analyzing all approaches we identified 10 categories from classifying the various analyses goals. Attention should be paid to the fact of multiple classification. For example, a security analysis is able to analyze dependencies and requirements and therefore can be assigned to both functional categories.

After we completed the functional classification, we conducted the technical categories. This procedure was more detailed and complex because of the large variety of existing methods in EA analysis. Only analyses with the same method and same steps of goal attainment can form a technical category. This constraint is necessary to enable discovery of shared characteristics. The already mentioned template was altered for creating technical categories. The new focus lies on the constructs, methods, techniques (including single steps) and artifacts. First, rough technical categories have been determined based on the dimensions "quantitative", "analytic", "simulation" and "functional" (Lankhorst, 2013). After this prestage detailed categories were created. Each identified analysis approach passed through this procedure. In contrast to functional categories every analysis was assigned to one specific technical category. As final result we concluded with 17 technical categories.

Altogether 105 analysis approaches fulfilled our

criteria and have been incorporated. Only 9 of them couldn't be classified. These ones were too specific and individual to create a category. For each analysis we identified exactly one technical category and at least one functional category. To create an overview of all possible combinations of categories three matrices were created. Two matrices represent the combination of the analysis approaches and the categories. Here we have to mention the features of both tables. The functional matrix has more possible combinations because analyses can achieve more targets simultaneously. However the technical matrix has only one combination per analysis. For example, Närman et al. (2008) is assigned to the functional categories **System Analyses**, **Attribute Analyses** and **Quality Analyses** and to the technical one **Bayesian Networks**. To provide an overview of the functional and technical combinations both matrices have been joined which resulted in a shared matrix (see figure 2). Thus, we get an overview of the realization techniques of a functional category and also the other way round for analyses and their used techniques. In the table an "x" represents that there is at least one analysis in current literature that was matched to both categories, the functional and the technical one.

3.2 Identification of Characteristics

As only properties and steps can show the components responsible for classification, we introduce characteristics to ensure accuracy. We define a characteristic as requirement, since an analysis can only be conducted target-aimed with all indispensable artifacts. Requirements support the achievement of goals and are used to identify hidden characteristics (Van Lamsweerde, 2001). Whereas properties can differ significantly, on some spots we had to choose the most elaborated one or to create a higher abstraction level. There are two types of characteristics: category specific ones and general characteristics. The second type includes a meta model and scenarios, determined at the beginning of an analysis. Another universal characteristic is the main goal. These characteristics have to be conducted for all analyses. Together they provide a high level of abstraction.

For the specific characteristics we distinguished five different kinds. The conducted kinds of characteristics are important for the identification of properties from technical analyses. Whereas functional categories have rough properties, technical categories have similar structure. We identified the following kind of characteristics: *Input*, *Conditions*, *Construct*, *Measurement*, and *Output*. The basis of an analysis is always represented in terms of *Input* data. In every

case an architecture or scenarios, in form of a model, are needed to conduct the following steps and final measurements. Before the main part of an analysis can be performed, sometimes *Conditions* are needed. For example the possibility of succeeding must be given. Most of the *Conditions* are analysis independent and therefore can be seen as generally valid. The main part and procedure of an analysis is the *Construct*, containing all details of the procedure. It's required to conduct all details successfully to be able to finish the analysis. Examples are detailed steps, mathematical algorithms, relationship types and weighting of artifacts. To prove and measure the results and development, every analysis needs some kind of *Measurement*. This characteristic is responsible to witness the achievement of goals. Most of the time a *Measurement* is proceeded with scales, KPIs and metrics to control functional and non-functional goals (Davis, 1989). This characteristic can vary dependent on the analysis and its goals. As last characteristic kind the *Output* was identified. It includes the way of presentation and type of outcome such as percentage, graphics or matrices. This category is crucial because analyses within the same category should have the same *Output*. We used these characteristic kinds to analyze the approaches again to receive detailed information. Through the new and detailed knowledge we had to rearrange the analysis categories on necessary points. New identified characteristics have been verified on correctness and necessity. After this step the final categorization of the analyses was received.

3.3 Functional Categorization

In the following we present the categories of the functional classification. Therefore we will list them combined with an example of a contained analysis approach. The assignment of all identified analyses to the categories can be found in (Rauscher, 2015). **System Analyses** (e.g. (Närman et al., 2008)) check partial or holistic systems. Mostly time quality aspects and their optimizations are in the main focus. Analyses which are contained in this category are often also part of other functional categories because of possible sub-goals. Specific attributes and their values are analyzed by **Attribute Analyses** (e.g. (Razavi et al., 2011)). The observation and management of attributes is the focus such approaches. For instance the different states of attributes with changing input can be analyzed. Analyses which prove dependent connections are classified as **Dependencies Analyses** (e.g. (Saat, 2010)). The main goal of these approaches is the identification of dependencies in EAs and connections of single components to receive

Functional \ Technical	Technical																	
	Bayesian Networks	Business Entities	PRM	Social Network	AHP	Time Evaluation	Tree	KPI and Matrices	Comparison	Views	Lifecycle	Ontology	EID	Weak Points	Matrices	Design	Structural	Other
System	x		x										x					
Attribute			x		x		x		x				x	x				
Dependencies	x		x				x		x		x	x			x			
Quality	x	x	x		x	x	x	x					x		x			x
Design				x					x			x			x	x	x	
Effects									x	x		x	x		x			
Requirements										x	x							
Financial								x						x	x			
Data													x					
Business Objects		x		x		x	x		x	x		x		x				
Other										x					x			x

Figure 2: Dependency matrix of the functional and technical categories.

an understanding of the whole architecture. **Quality Analyses** have the main focus on various quality questions regarding attributes, systems, architectures and other components and target subjective and measurable goals (e.g. (Närman et al., 2008)). This category is based on ISO 9126 and analyzes, for instance, maintainability, maturity and interoperability. Another category represents the analysis of architecture design (**Design Analyses**), examples are (Aier et al., 2011; Kazienko et al., 2011). Through receiving an overview on the architecture construct all possibilities of designs can be identified. Beside the analysis of holistic or partial design, business entities, procedures and components can be analyzed and used to optimize the architecture. All approaches which control impacts in architectures and actions are joint in **Effect Analyses** (e.g. (de Boer et al., 2005)). In contrast to **Dependencies Analyses** these approaches observe the direct impact and effects of changes in architecture elements. **Requirements Analyses** target all possible requirements to achieve states or goals (e.g. (Aier et al., 2009)). To accomplish special goals, different requirements are needed. Results are either specified values or features and entities. To identify costs and benefits **Financial Analyses** are used (e.g. (Niemann, 2006)). On top financial weak points and possible impacts can be discovered. These analyses present a measurement with mathematical calculations. Therefore key figures and metrics are able to evaluate outcomes. However receiving affected entities is a side effect of the results. Consequently **Financial Analyses** observe too high costs or uncertainty and hence are an indicator of necessary architecture and procedure changes. However **Data Anal-**

yses cover all kinds of data (e.g. (Närman et al., 2009)). The focus lies on quality and accuracy, because data are crucial for successful EA operations. Finally the category **Business Object Analyses** was identified (e.g. (Della Bordella et al., 2011)). Business objects of every kind, e.g. operations, artifacts and entities, which are part of architecture procedures and construct are addressed here.

3.4 Technical Categories

In the following we describe the 17 technical categories. For the description we chose the most important and marked characteristic kinds (see section 3.2). Since nearly all technical categories require an architecture model, scenarios and goals as *Input* we won't mention it below. The first technical category represent analyses conducted with **Bayesian Networks** (e.g. (Närman et al., 2008)). Analyses of this category use this technique to analyze the quality. It is reused in other analyses as part of their procedures, e.g. PRM analyses. Firstly a model with Bayesian Networks is built in the main *Construct*, including all nodes and connections of the architecture. A node represents a variable with a conditional probability distribution. Therefore in the next step probabilities of attributes and the whole model can be calculated while creating matrices with discrete ranges. In conclusion this category has probability values as *Output* and can answer questions about the probability of an attribute's status. **Business Entities** are a method to receive artifacts and analyze quality (e.g. (Della Bordella et al., 2011)). As *Input* and *Conditions* the goal type, strategies and quality features

must be determined. First step of the *Construct* detects advantages, operations and elements of strategies. As a second step, influencer and strategies are combined to observe the goals. For measurement the goal is quantified and evaluated. The *Output* contains valued strategies, quality values and identified operations and entities. **Probabilistic Relational Models (PRM)** contains 14 analyses and is therewith the most used technique (e.g. (Buschle et al., 2011)). For instance dependency and quality analyses can be conducted with PRM. *Conditions* require determinable goals and controllable attributes. As a prestep of the *Construct* connections are defined and uncertainties are formalized. Hereafter the PRM is used to calculate the conditional probability of all scenarios and of the dependencies and attributes. Therefore percentage of attributes, scenarios and uncertainty are in the *Output*. **Social Network** analyses differ deeply from the other categories (e.g. (Kazienko et al., 2011)). To conduct the analysis questionnaires and all available documents, like bills and connections are used. For the *Input* all available nodes (= entities) and connections are required. As *Construct* clusters are built and properties can be checked. Additional new entities and connections are found. For the *Measurement* a matrix with entities and factors is created for quantitative evaluation with factors or for identification of weak points. An overview of the whole architecture construct and its entities and connections on a social basis is found in the *Output*. **Analytic Hierarchy Process (AHP)** can be used for analyzing attributes and quality aspects and is one of the most elaborated EA analysis techniques (e.g. (Razavi et al., 2011)). *Conditions* are experts knowledge used for weighing as well as definable quality attributes and level of success. The *Construct* determines quality attributes, their criteria, subcriteria, layers and level of importance. To evaluate the attributes, criteria and layers pairwise comparisons are conducted by experts. For every part which passes the comparison process a prioritized vector is created. Afterwards the same proceeding is conducted for all scenarios. This results in an *Output* with prioritized lists of quality attributes and scenarios to determine the best scenario dependent on the attribute. The method of **Time Evaluation** (e.g. (Lankhorst, 2013)) calculates times and observes the quality of business entities and operations. As additional information the trigger and arrival times are required for the *Input*. Rules (*Condition*) are necessary to cut the architecture in views and receive five perspectives with single time measurements. In this category *Construct* and *Measurement* are combined. Within the views the specific times are calculated. Examples are ‘Costumer View’

and ‘Process View’ with ‘Processing Time’ and ‘Response Time’. In a last step, all calculated times are summed up to a total time. **Trees** are used to analyze and identify dependencies, coherences and quality features. The *Output* of those analyses delivers the probability of an occurring failure or specific quality, depending on the purpose. In the beginning of the *Construct* the goals, entities and relations are defined. Afterwards a fault tree is built using Bayesian Networks, containing all steps or events required for the procedure of operation. For every component of this tree a conditional probability matrix is created to receive the probability of failures or quality property (Närman et al., 2011). The usage of **KPI** (Key Performance Indicator) is used in most analyses with quantitative measurement. Because of the high variety of contained analyses, a high level of abstraction was used. The goal has to be definable definite, measurable, agreed, realistic and time-bound. The *Construct* starts with identifying all artifacts which should be analyzed and determining the matching KPIs. The analysis evaluates the artifacts and compares the values with metrics for *Measurement*. The result can present achieved goals, unsatisfied quality issues and the financial situation (Niemann, 2006). **Comparison** is a simple but powerful method (e.g. (de Boer et al., 2005)). Next to whole alternatives, also single scenarios, processes, attributes and dependencies can be compared to against each other. To choose the best option, the as-is and to-be state should exist. As first step of the *Construct* a model is created containing all components which should be analyzed. Afterwards the models are compared with a previous state or with another alternative. On this way all possible states can be observed and the best alternative to achieve the goals is identified. The technique **Views** is often only a part of another analysis. However we identified analyses having highly elaborated approaches with the main focus on views (e.g. (Sasa and Krisper, 2011)). For instance views analyze effects and requirements. Therefore criteria and their desirable perspective have to be specified. After this the views can be built with all required components. A definite *Measurement* is not contained in this category. However, views can be evaluated with criteria to observe whether the view can achieve its goals, for example the processing time. A less popular methodology is the observation of **Lifecycle** (e.g. (Saat, 2010; Aier et al., 2009)). These analyses ascertain requirements and dependencies through consideration of architecture phases. In this way changes are identified and it’s possible to determine whether an artifact presents a specific state at an specific time. To conduct this, lifecycle phases, attributes, times and states are nec-

essary as *Input*. The analysis *Construct* displays the cycle on analyzed areas. Afterwards for every product the state is checked at a special point. For *Measurement* probabilities are calculated. These results, with the changed cycle are displayed in the *Output*. **Ontology** is a typical technique of EAM, however uncommon in analysis (e.g. (Sunkle et al., 2013)). A special meta model created with ontologies and ontology rules is required for the *Input* and *Conditions*. The *Construct* analyzes entities, resources, persons and products, determines the ontology of entities and defines connection types. Afterwards dependencies, viewpoints and special factors can be evaluated for the outcome. **EID** (Extended Influence Diagrams) are the third most used technique to conduct analysis in EA (e.g. (Johnson et al., 2007)). Possible results can be statements about maintainability, security and availability. Therefore systems, attributes, quality, impacts and data are analyzed. As *Condition* it has to be ensured that all contained components are able to be built with EID. Afterwards all scenarios, goals and entities are represented as EID nodes and connections. For *Measurement* Bayesian Networks are used to calculate the probabilities of attributes. Thus it's possible to analyze dependencies by inferring changes and altered values. Another identified technique is the usage of a **Matrices** (e.g. (Szyszka, 2009)). Matrices can be used in various ways, mostly they are utilized to present results. Therefore dimensions and the kind of measurement are determined in order to build and evaluate the matrix. The results can vary from quantitative outcomes to weak points, redundant artifacts and functional dependencies. Analyses joint in the category **Design** are only able to observe architecture design in a specific way (e.g. (Aier et al., 2011)). As *Condition* factors and expert knowledge is needed. In the main *Construct* items and data are determined, factors are checked with questionnaires and a cluster analysis is conducted. As *Measurement* a matrix of items and factors is built and evaluated. The results of the evaluation represent the *Output*. For the identification of **Weak Points** (e.g. (Xie et al., 2008)) and their costs the following procedure can be used. Resources are needed to create a matrix of workflows, resources and their availabilities. This matrix is filled with connections and their values and is the basis for availability calculations. Whenever the availability is higher as the respective requirement, the condition is fulfilled. In addition it's possible to weight resources and receive alternatives with higher availability. The last technical category contains analyses with **Structural** procedure (e.g. (Buckl et al., 2009a)). This analysis tries to observe design through displaying obstacles of different architecture versions. Therefore a docu-

mentation is needed as *Condition* and the main part of analysis consist of an observation of changes. The *Output* type is unique and represents potential obstacles caused by different versions.

4 EVALUATION

The identified requirements for the 10 functional categories and the 17 technical categories are formalized using a domain specific language in order to validate the categorization. Therewith we can elaborate the integrity and correctness of the requirements, i.e. if they are sufficient to describe the analyses in an adequate way. For each category we chose an analysis approach from literature and formalized it using the developed language. In the following we present the DSL and illustrate its usage. Finally we provide a short discussion about our results.

4.1 DSL for Analysis Description

For the language development we used Xtext¹, a framework that comprises a powerful language for the description of textual languages. The model is generated by the framework as well as an parser, linker, type checker and compiler. The DSL was developed according to the meta model development process for abstract syntax development in (Brambilla et al., 2012). This incremental and iterative process consists of three phases: The 'Modeling Domain Analysis' phase, elaborating the purpose and content, the 'Modeling Language Design' phase, defining the meta model, and the 'Modeling Language Validation' phase, verifying the correctness and integrity. For the last step we select representative EA analyses for each category and formalize them using our modeling language. Difficulties and mistakes during modeling triggers a new iteration of the development process. The concrete syntax is developed simultaneously with the abstract syntax due to the nature of Xtext.

The developed DSL is structured in a general and in a categorization specific part. Figure 3 shows the main rule for the analysis language and the realization of the dimensions. General requirements that occur in all categories are summarized at beginning in the main rule. This is the name of the analysis, the required meta model and possible scenarios to evaluate. For description of the meta model and the scenarios we developed a language construct that allows to specify them similar to UML. The goal of an analysis can be modeled using a string and its type is

¹Xtext <https://eclipse.org/Xtext/index.html>

```

MetaLanguage:
'EAM Analysis Language' '{}'
//Domain Definition: General Requirements
'Performing Analysis' analysis=STRING
'Metamodel' model+=UMLModel ('{')
'Scenarios:' scenarioName+=NameIdentifier (scenarioModel+=UMLModel)*
(';" scenarioName+=NameIdentifier (scenarioModel+=UMLModel)*')
'}'?
'Goal' goal=STRING
'Goal Type':' goalType+=GoalType ('&' goalType+=GoalType )*
'}'
//Choice of Dimensions
('Category functional Dimension' ':' functional+=Functional)?
('Category technical Dimension' ':' technical+=Technical)? ;

//-----Functional Categories-----//
Functional:
SystemAnalysis | AttributeAnalysis | ... | BusinessObjectAnalysis;

//-----Technical Categories-----//
Technical:
BayesianNetworks | BusinessEntities | ... | Structural ;

//Choice of a possible technique matching to the chosen functional Category
SystemAnalysis:
'System Analysis' (':')?
('Technique' analysisTechnique+=SystemAnalysisTechnique)? ;

SystemAnalysisTechnique:
EID | PRM | BayesianNetworks ;
    
```

Figure 3: DSL for EA analyses - main rule.

defined using an enumeration. Possible goal types are: percentage, matrix, probability, dependency, object, effect, scenario, number or boolean. The two-dimensional categorization is realized as followed: The user can either first choose the functional dimension and then the technical, but also the other way round. The rule system of the DSL restricts the second dimension to those that are available. For example the functional dimension **System Analysis** has realizations with the technical dimensions **EID**, **PRM** and **Bayesian Networks**. The rule *SystemAnalysis-Technique* ensures the integrity of the selection according to the matrix (figure 2).

```

EAM Analysis Language{
Performing Analysis "Information Security Analysis"
Metamodel Model "Architecture of Information Security"{
Class "Application"{ ... }
...
}
}Scenarios:
"Scenario 1" Model"UML Model Scenario 1"{ ... };;
"Scenario 2" Model"UML Model Scenario 2"{ ... }
}
Goal"Probability of quality attributes for security"
Goal Type :Percentage
}
Category functional Dimension:Attribute Analysis:
Technique Extended Influence Diagram
INPUT{
Metamodel "Architecture of Information Security"{
Scenario"Scenario 1",
Scenario"Scenario 2"
}
}
}CONSTRUCT{
EID MODEL ELEMENTS{
Chosen Scenario "Scenario 1"
//Value assumptions
Scenario Node:
type: Decision Node "Scenario Selection" Value: 0."90"
Goal: type: Utility Node "Profit" Value: 0."0"
Attributes:
type: Chance Node"User Training Process" Value: 0."75"
...
Relations:
"Scenario Selection" as Causal Relation to "User Training Process"
}...
}
}
    
```

Figure 4: Excerpt of the description of EID analysis using the DSL.

For each technical category a rule is implemented that satisfies the requirements specified in chapter 3.

The rules comprise statements for defining the input, the conditions and construct, the measurement and the output. To illustrate the structure of a category definition figure 4 shows an example description of the Information Security Analysis from (Johnson et al., 2007). This analysis evaluates the architecture by calculating the probability of quality attributes for security. Corresponding to the main rule the description starts with the analysis name followed by a specification of the meta model and two scenarios. The meta model describes the classes, relationships and attributes that are necessary for the analysis. The two scenarios represent different alternatives that should be evaluated. The scenario description is followed by the goal statement and the goal type, in this case *percentage*. Then the functional and technical dimension are defined. The functional dimension is **Attribute Analysis** and the technical one is **Extended Influence Diagram**. Following the remaining structure of the analysis specification is specific for analyses of the category **EID**. The input of the analysis is here straightforward the defined meta model and both scenarios. The construct part defines the requirements in order to create extended influence diagrams. First the chosen scenario is set, then the nodes, goals and attributes with their types and values are defined. Finally the EID specific relations are specified.

4.2 Discussion

We were able to define and apply a domain specific language for the description of EA analyses. The identified categories, functional as well as technical, are integrated. It was also possible to describe the analysis approaches from literature using the DSL. Additionally the language is easy extendable and without special knowledge understand- and usable. The language can be reused for the development of new analyses, since it provides a sound foundation of requirements that have to be extended. After further development the language can also be used as an entry point for the specification and execution of analysis. Most of the requirements for the technical categories are realized. A few requirements are determined as given and not further mentioned in the language, since these requirements are obviously. Examples are the possibility to raise data or whether data can be used and be accessible. In addition requirements which are not verifiable couldn't be included. For instance, it is not verifiable whether the meta model is usable to achieve goals, whether artifacts are able to image with EID components or used nodes are controllable. Additionally the acceptance of an used technique or the availability of experts knowledge is not verifiable and

thus not integrated in the language. Requirements that are defined in a graphical way, for example matrices, are difficult to realize in a textual language. Also the definition of patterns is only specified with limitations in the language.

The lower number of functional categories in contrast to the technical one can be explained with the focus on one field of interest. Since we concentrate on pure EA analyses the analysis goals were repetitive. A problem during categorization was the issue that not all aspects from the analysis are described in detail in the available publications. At this point we were only able to identify limited requirements or we had to make assumptions to proceed. A interrelated problem is the fact of the low amount of available descriptions of conducted analyses to evaluate our language. Additionally some analyses use very specific techniques or modeling approaches. Here, it was not possible to consider all details in order to create a sound categorization. We abstracted from some specifics in order to define the general requirements for a category. We received generally valid requirements by orienting on the approach which is most elaborated and create a higher level of abstraction. An example is the technical category KPI with a high abstraction level. The contained analyses differ deeply in measuring values with different formulas.

Encountered categorization approaches in related work tried to work on a meta level after studying the analyses. However, in contrast to our target they designed an analysis framework meta model independently (Langermeier et al., 2014), developed a category independent meta language (Buckl et al., 2011) and had the main focus on characteristics (Buckl et al., 2010). Additional EA analyses can be distinguished between the point of execution time. Therefore the analyses are sorted in ex-ante and ex-post to determine whether an analysis will be conducted before or after the adoption of an architecture. It is also possible to separate the analyses according to their execution technique: expert-based, rule-based or indicator-based (Buckl et al., 2009b). However, both classifications are not detailed enough to identify characteristics and most of analyses can't be strictly classified within these divisions. Lankhorst (2013) conducted an initial categorization with the already mentioned four dimensions. Quantitative and functional differ at the input and output data, whereas functional can be distinguished in static or dynamic. However this division is not detailed enough to identify the explicit requirements of classified analyses and four categories is a rough classification.

5 CONCLUSION

In this paper we presented a two-dimensional categorization of EA analyses, based on the characteristics of the approaches found in literature. The first dimension addresses the functional aspect, the second one the technical aspect. Altogether we identified 105 analyses, which are classified in 10 functional categories and 17 technical categories. Using this categorization we can identify 40 different analysis types used in EA. The dependencies between the approaches of the functional and technical dimensions are visualized in a matrix. The dependencies as well as the characteristics of the analysis categories are formalized with a domain specific language. The language provides a structural way to represent the preconditions of an analysis, the technical requirements for execution and also the outcome of it. Beside evaluation purposes the language can also be used by the enterprise architect to decide whether the outcome of an analysis is from interest for his question, if the analysis is applicable on his EA model and how great the effort of adaption are, in order to execute the analysis. The idea of such an EA analysis catalog is the support of reuse of existing work in the domain of EA. Therefore future work has to investigate techniques for context independent execution of those analysis. This could be the development of tool support for the usage of the categories and the DSL. Thus computations which need further tools can be included, new analyses could be created easily and requirements are checked automatically. Additionally a higher abstraction level of the category characteristics would be conceivable to make the requirements general valid.

REFERENCES

- Aier, S., Buckl, S., Franke, U., Gleichauf, B., Johnson, P., Närman, P., Schweda, C. M., and Ullberg, J. (2009). A survival analysis of application life spans based on enterprise architecture models. In *3rd Workshop on EMISA*, pages 141–154.
- Aier, S., Gleichauf, B., and Winter, R. (2011). Understanding enterprise architecture management design-an empirical analysis. In *Proceedings of 10th Conference on Wirtschaftsinformatik*.
- Brambilla, M., Cabot, J., and Wimmer, M. (2012). *Model-driven software engineering in practice*.
- Buckl, S., Buschle, M., Johnson, P., Matthes, F., and Schweda, C. M. (2011). A meta-language for enterprise architecture analysis. In *Enterprise, Business-Process and Information Systems Modeling*, pages 511–525. Springer.

- Buckl, S., Matthes, F., Neubert, C., and Schweda, C. M. (2009a). A wiki-based approach to enterprise architecture documentation and analysis. In *ECIS 2009 Proceedings*, pages 1476–1487.
- Buckl, S., Matthes, F., and Schweda, C. M. (2009b). Classifying enterprise architecture analysis approaches. In *Enterprise Interoperability*, pages 66–79. Springer.
- Buckl, S., Matthes, F., and Schweda, C. M. (2010). A Meta-language for EA Information Modeling State-of-the-Art and Requirements Elicitation. In *Enterprise, Business-Process and Information Systems Modeling*, pages 169–181. Springer.
- Buschle, M., Ullberg, J., Franke, U., Lagerström, R., and Sommestad, T. (2011). A tool for enterprise architecture analysis using the PRM formalism. In *Information Systems Evolution*, pages 108–121. Springer.
- Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, 13(3):319–340.
- de Boer, F. S., Bonsangue, M. M., Groenewegen, L., Stam, A., Stevens, S., and Van Der Torre, L. (2005). Change impact analysis of enterprise architectures. In *IEEE International Conf. on Information Reuse and Integration*, pages 177–181.
- Della Bordella, M., Liu, R., Ravarini, A., Wu, F. Y., and Nigam, A. (2011). Towards a method for realizing sustained competitive advantage through business entity analysis. In *Enterprise, Business-Process and Information Systems Modeling*, pages 216–230. Springer.
- Johnson, P., Lagerström, R., Närman, P., and Simonsson, M. (2007). Enterprise architecture analysis with extended influence diagrams. *Information Systems Frontiers*, 9(2):163–180.
- Kazienko, P., Michalski, R., and Palus, S. (2011). Social network analysis as a tool for improving enterprise architecture. In *Agent and Multi-Agent Systems: Technologies and Applications*, pages 651–660. Springer.
- Krcmar, H. (2015). *Informationsmanagement*. Gabler.
- Langermeier, M., Saad, C., and Bauer, B. (2014). A unified framework for enterprise architecture analysis. In *18th IEEE International EDOC Conference Workshops*, pages 227–236.
- Lankhorst, M. (2013). *Enterprise Architecture at Work: Modelling, Communication and Analysis*. Springer.
- Närman, P., Franke, U., König, J., Buschle, M., and Ekstedt, M. (2011). Enterprise architecture availability analysis using fault trees and stakeholder interviews. *Enterprise Information Systems*, 8(1):1–25.
- Närman, P., Johnson, P., Ekstedt, M., Chenine, M., and König, J. (2009). Enterprise architecture analysis for data accuracy assessments. In *13th IEEE International EDOC Conference*, pages 24–33.
- Närman, P., Schonherr, M., Johnson, P., Ekstedt, M., and Chenine, M. (2008). Using enterprise architecture models for system quality analysis. In *12th IEEE International EDOC Conference*, pages 14–23.
- Niemann, K. D. (2006). *From enterprise architecture to IT governance*. Springer.
- Rauscher, J. (2013). Analysen in Unternehmensarchitekturen - Ziele, Techniken, Anwendungsbereiche. *Bachelor Thesis, University Augsburg*.
- Rauscher, J. (2015). Anforderungen an und Definition von einer Analysesprache für das Enterprise Architecture Management. *Bachelor Thesis, University Augsburg*.
- Razavi, M., Aliee, F. S., and Badie, K. (2011). An AHP-based approach toward enterprise architecture analysis based on enterprise architecture quality attributes. *Knowledge and information systems*, 28(2):449–472.
- Saat, J. (2010). Zeitbezogene Abhängigkeitsanalysen der Unternehmensarchitektur. In *MKWI*, pages 119–130.
- Sasa, A. and Krisper, M. (2011). Enterprise architecture patterns for business process support analysis. *Journal of Systems and Software*, 84(9):1480–1506.
- Sunkle, S., Kulkarni, V., and Roychoudhury, S. (2013). Analyzable enterprise models using ontology. In *CAiSE Forum*, volume 998, pages 33–40.
- Szyszka, B. (2009). Analysis and classification of maturity models in enterprise architecture management. *Bachelor Thesis, Technical University Munich*.
- Van Lamsweerde, A. (2001). Goal-oriented requirements engineering: A guided tour. In *5th IEEE International Symp. on Requirements Engineering*, pages 249–262.
- Wan, H. and Carlsson, S. (2012). Towards an understanding of enterprise architecture analysis activities. In *Proceedings of 6th ECIME*.
- Winter, R. and Fischer, R. (2006). Essential layers, artifacts, and dependencies of enterprise architecture. In *10th IEEE International EDOC Conference Workshops*.
- Xie, L., Luo, J., Qiu, J., Pershing, J., Li, Y., Chen, Y., et al. (2008). Availability “weak point” analysis over an SOA deployment framework. In *IEEE Network Operations and Management Symposium*, pages 473–480.
- Zia, M. J., Azam, F., and Allauddin, M. (2011). A survey of enterprise architecture analysis using multi criteria decision making models. In *Intelligent Computing and Information Science*, pages 631–637. Springer.

Visualization of Changes in ArchiMate

Within the Application and Technology Layers

Robert Bakelaar¹ and Ella Roubtsova²

¹Royal Vopak, Global IT, Westerlaan 10, 3016CK, Rotterdam, The Netherlands

²Open Universiteit of The Netherlands, Valkenburgerweg 177, 6419 AT Heerlen, The Netherlands
robert.bakelaar@vopak.com, ella.roubtsova@ieee.org

Keywords: Changes of Enterprise Architecture, ArchiMate, Visualisation of Changes, Best of Breed.

Abstract: Enterprise architecture is periodically changed. The visualization of changes is demanded for communication of the teams implementing changes. In this paper, we use the modern cases of transformation of ERP (Enterprise Resource Planning) systems to the Best of Breed solutions and the popular modeling language ArchiMate with its extensions to propose a method for visualization of changes. The method covers changes within the application and technology layers and their link to the business layer. It includes artifacts, principles and means for visualization. The method is tested with real cases of changes of ERP using the Best of Breed strategy.

1 INTRODUCTION

Visualization of changes in enterprise architecture languages is not a simple task as such languages reflect the static structure of an enterprise. However, enterprise architecture is periodically changed. Changes are usually implemented by several teams. For communication of teams, the changes need to be visualised.

In this paper, we use the modern cases of changes of ERP (Enterprise Resource Planning) systems using the Best of Breed strategy and the popular modeling language ArchiMate (The Open Group, 2013) to show the needs for change visualization and to explore the difficulties during the visualization process and abstractions needed for it. As a result of our exploration, we propose a visualization method and visualization means.

We have chosen the cases of changes from ERP to Best of Breed because these cases are representative enough to cover different modifications within the application and technology layers.

Our choice of the ArchiMate modeling language is explained by several reasons.

- Firstly, ArchiMate is related to the TOGAF (the abbreviation of “The Open Group Architecture Framework”). It is a standard for enterprise architecture (The Open Group, 2016).
- Secondly, ArchiMate is based on a metamodel

that contains a layered structure in which the business, application and technology architectures are covered and the modeling of relations between the different layers is supported. Having means to model different domains of the same business, ArchiMate fills the gap between the different domain architectures and the missing relationships between these architectures, as they are most of the time created by separate architects in different modeling languages (Lankhorst et al., 2009).

- Thirdly, the ArchiMate language is used in industry, and it is supported with some open source tools, for example, Archi 2.4. (Institute of Educational Cybernetics, 2012)

We investigate the ArchiMate and its extensions. As a result, we propose a visualization methodology based on the ArchiMate and its extensions and motivate our suggestions with the cases of transformation of ERP using the Best of Breed strategy.

The structure of the paper is the following:

Section 2 describes some characteristics of ArchiMate relevant to the visualization of changes.

Section 3 provides information about our experimental approach for development of the visualization methodology in ArchiMate.

Section 4 presents our method for visualization of changes in ArchiMate and its known extensions.

Section 5 discusses the principles, means and abstractions used for visualization and the scalability is-

sues.

Section 6 concludes the paper.

2 ARCHIMATE

ArchiMate is one of the popular tools for visualization of enterprise architecture (Fritscher and Pigneur, 2011), (Lankhorst et al., 2009). Mostly it is because it supports visualization of three related layers of the enterprise architecture:

1. the business layer (products, services, actors, processes);
2. the application layer (application components, application functions and data objects) and
3. the technological layer (infrastructure services, hardware, system software).

There are also ArchiMate extensions (The Open Group, 2013) that cover high level concepts used in business implementation. Among them are the motivation extension and the implementation and migration extension.

The visual elements of ArchiMate within the application and technology layers are mostly concepts and relations (Lankhorst et al., 2010):

- Active elements performing behavior (i.e. application service, application component).
- Elements describing behavior (i.e. infrastructure service, application function).
- Passive elements on which behavior is performed (i.e. data object).
- Relations between the elements, depicted as connecting lines between elements or boxes.
- The visual elements can be included into other elements by aggregation, composition and grouping relations.

In the motivation, implementation and migration extensions, there are:

- Elements describing motivation: goals, requirements, principles;
- Elements describing migration: plateaus and gaps.

The elements used in this paper are listed in Figure 1.

Our research questions were the following. Are the visual elements of ArchiMate and its extensions sufficient to visualize changes of architecture? What is a method of application of visual elements for visualization of architecture changes?

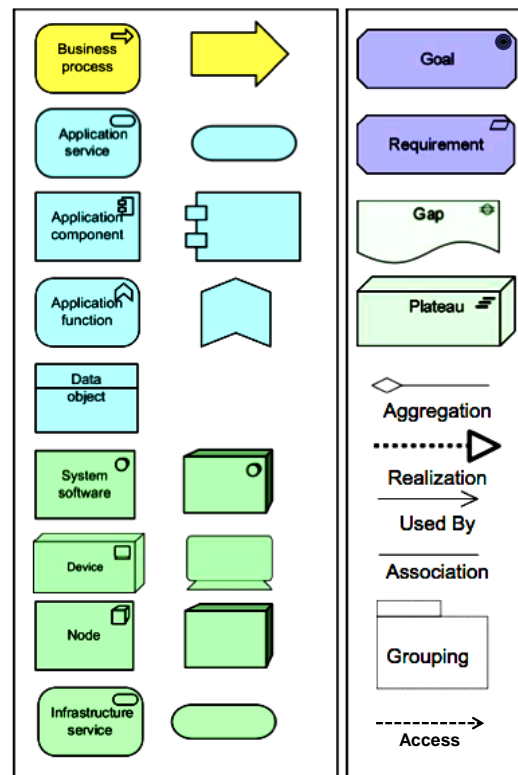


Figure 1: Visual elements of ArchiMate and its extensions used in this paper

3 DEVELOPMENT OF A VISUALIZATION METHOD

We apply the experimental approach to propose a visualization method for changes of an enterprise architecture. The experiment is set on the ERP implementation within Royal Vopak (<https://www.vopak.com/>) modified using the Best of Breed strategy.

3.1 Best of Breed

Business processes often contain some parts of functionality that are well supported by standard solutions. The Best of Breed strategy is directed to use the most suitable standard software and to develop only the parts that are not supported by standard software (Light et al., 2001). This approach promises flexible application implementation, low costs of maintenance and changes (Cardoso et al., 2004).

Many companies have ERP systems that provide all the applications for an enterprise and integrate them in a superior solution where every module may not be the best of its class. If a company has an imple-

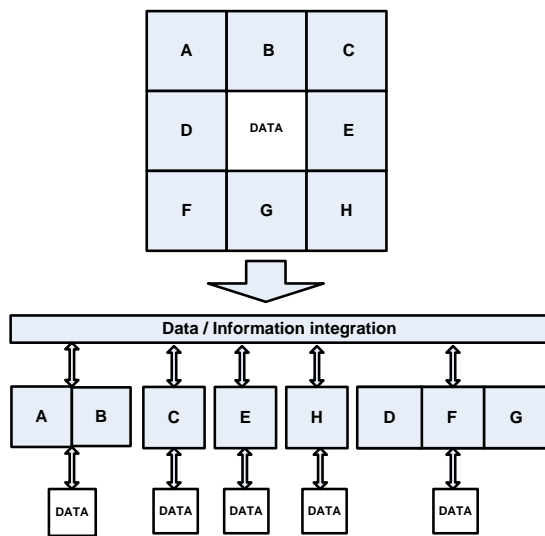


Figure 2: From ERP to Best of Breed: The ERP implementation of functions (A, B, C, D, E, F, G, H) are decomposed into the best applications available on the market (A, B) , (D, F, G) , C, E and H .

mented ERP system and wants to change it in order to use the best standard solutions, the company needs to visualize, examine and change its architecture.

Ideally, the transformation is the decomposition of the application functionality and the data, as it is shown in Figure 2. In reality, the elements are not only decomposed, but also removed, added and changed. Because of that, we have studied several cases of architectural changes.

3.2 Cases of changes

We consider the following business situations that cover the transformation from ERP to Best of Breed:

1. The business process remains unchanged, but the changes should be made in the application and the technological layers by adding, changing and removing elements.
2. The relations between applications may change.
3. The relations between technological elements and applications may change.
4. The relations between business functions and applications may change.

For each of the situations, we analyse what should be visualised. Doing the experiment, we have discovered the repeated steps, artifacts, principles and means useful for visualization of changes in ArchiMate and also some missing relations that are needed to visualise changes.

4 VISUAL ARTIFACTS

The set of visual artifacts depends on the chosen strategy and on the given requirements. Therefore, we need a goal/requirements view.

Our strategy is the Best of Breed, where the decomposition of existing application functions, new functions and obsolete functions should be shown on architectural views.

Then the question of the technological support of application functions should be solved. The key difference between ERP and Best of Breed is the way of data/information integration via a service bus (Figure 2). This means, that at least one view should examine the communication between application functions.

At the end, it is good to remind that the changes are implemented for business. Therefore, the relations of the new and decomposed functions with the business layer should be visualized.

In all cases of the transformation from ERP to the Best of Breed solutions, we have found that visualization of changes consists of visual artifacts produced in the order presented below.

1. Strategy of changes; business strategy, IT principles and requirements - the motivation behind the IT transformation.
2. Transition between the As-Is and To-Be architecture, including the gap of changes and the goals, requirements and principles related to the architecture states and changes.
3. As-Is and To-Be architectures at the application layer without specific indication of changes.
4. Gap of changes, presented at the application layer. The focus is on changed elements and on relations between the obsolete and new elements.
5. Gap of changes, combining the application and technology layers and the motivation.
6. Gap of changes at the application and technology layers with the focus on communication elements.
7. Gap of changes, at the application and business layers with the focus on relations between business steps and the applications supporting these business steps.

4.1 Illustration of Visual Artifacts

We illustrate the proposed visualisation method with one of our cases “Replacement of an Order Management Component in the Order Management and Invoicing service”.

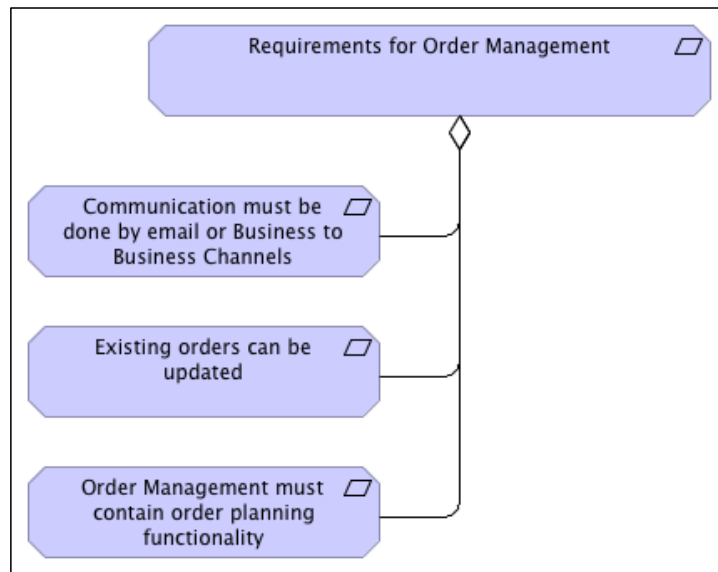


Figure 3: Example of Artifact 1: Requirements for changes.

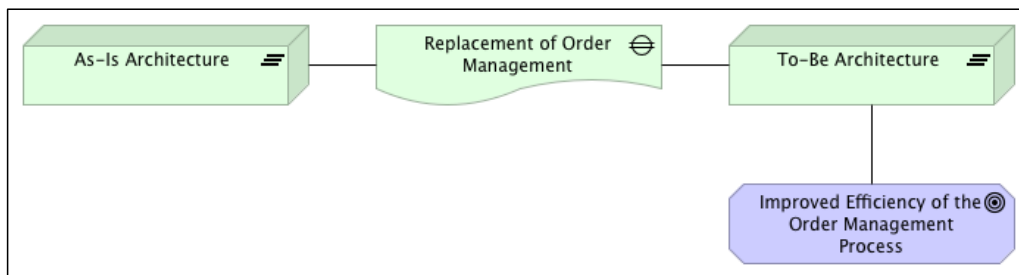


Figure 4: Example of Artifact 2: Abstract transition between As-Is and To-Be architectures.

4.1.1 Strategy

Figure 3 shows an example of the visual artifact (1), the requirements for changes. In the To-Be architecture, the orders should be taken via e-mail and B2B channels. The functions of validation and updating of the taken orders should be preserved. The motivation extension of ArchiMate is used for visualization of requirements in Figure 3.

4.1.2 Transition

The migration extension is used to draw the transition from one architecture to another (artifact (2)). Plateaus are used to represent As-Is and To-Be architectures and a gap visualizes the actual changes. Figure 4 represents the transition between As-Is and To-Be.

4.1.3 As-Is, T-Be

Figures 5 and 6 represent the visual artifacts (3), the

As-Is and To-Be architectures at the application level. As the reader may see, the time consuming and error prone comparison of the As-IS and To-Be architectures should be done in order to find the changes. So, to ensure the understanding of changes, the views on the gap of changes should be provided.

4.1.4 Views on a Gap of Changes

In the case of decomposition of application functions (Best of Breed strategy) with new functions, we need to focus first on application layer, application functions and on relations between the obsolete and and new functions. Figure 7 is an example of the artifact (4), being a view on the gap of changes with the focus on application functions and on the relation between the obsolete and new application functions. Figure 7 shows that the set of standard ArchiMate elements should be extended with new relations *<Extended-by>*, *<Replaced-by>* to enable visualization of relations between the obsolete and new elements.

The Best of Breed strategy includes changes

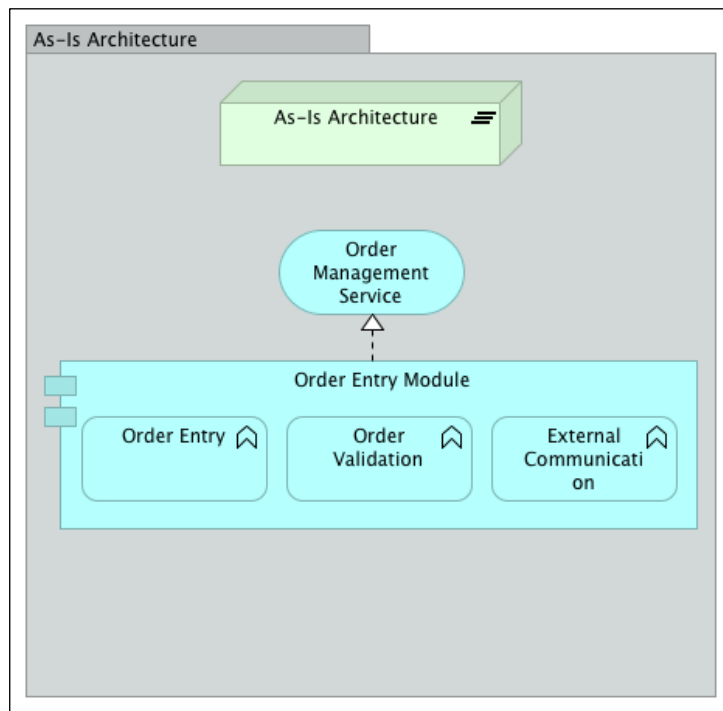


Figure 5: Example of Artifact 3: As-Is Architecture.

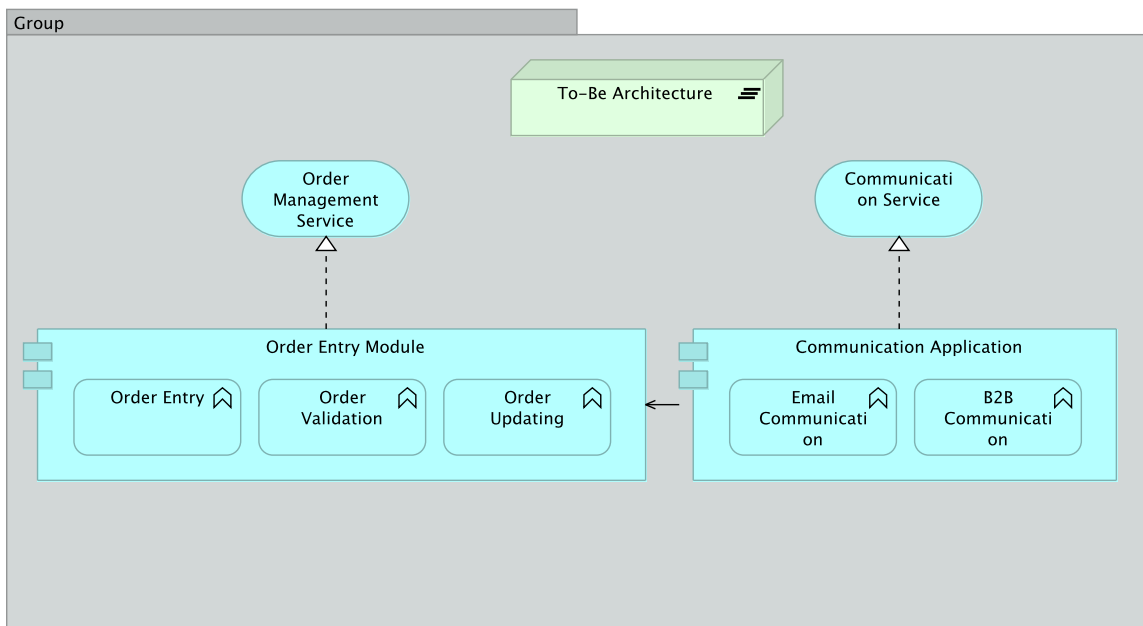


Figure 6: Example of Artifact 3: To-Be Architecture.

within the technological layer, because the implemented applications are taken from the market. Therefore, the artifact (5) (Figure 8) presents the application and technology layers with motivation and still keeps the focus on changes. It shows that the

messaging service, implemented on the SaaS platform, has been chosen for the e-mail and B2B order taking.

Because the Best of Breed application functions are differently organized, the communication of them

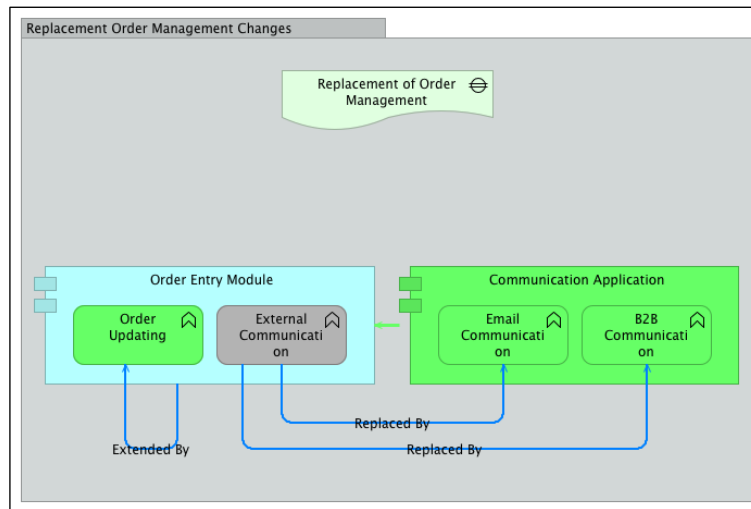


Figure 7: Example of Artifact 4: Gap of changes with focus on relations between the old and new application functions.

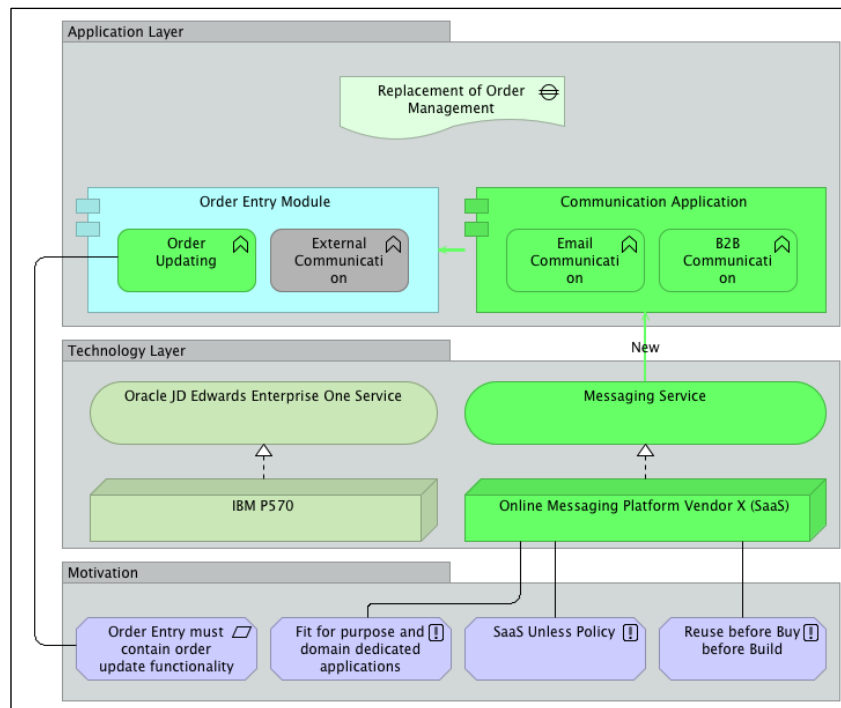


Figure 8: Example of Artifact 5: Gap of changes with focus on relations between the application, technology and motivation layers.

is also changed. The next artifact (6) has the focus on communication. Figure 9 presents the gap of changes, but abstracted from the application functions and focused on the communication elements. Two groups of application functions implement two services: the *Communication Service* and the *Order Management Service*. The *Order Management Service* publishes messages to the Enterprise Service Bus. The *Commu-*

nication Service subscribes for these messages.

Figure 10 shows the artifact (7), i.e. the relations of the application layer and the business layer. We can see that the process step *Contact relevant party* of the process *Order Intake, planning and scheduling* is supported by the *Communication Application*.

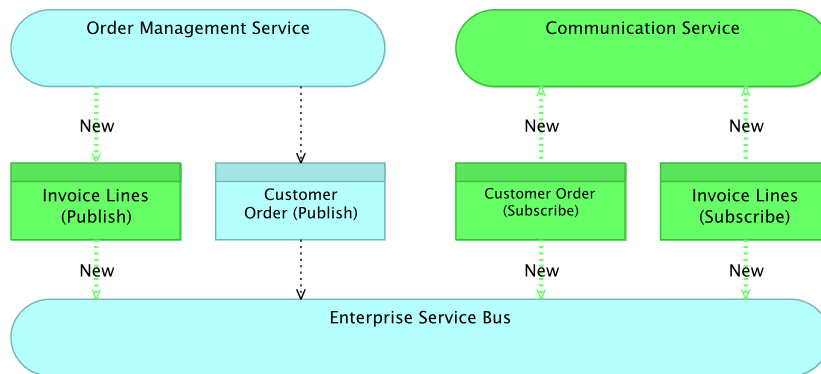


Figure 9: Example of Artifact 6: Gap of changes with focus on communication elements.

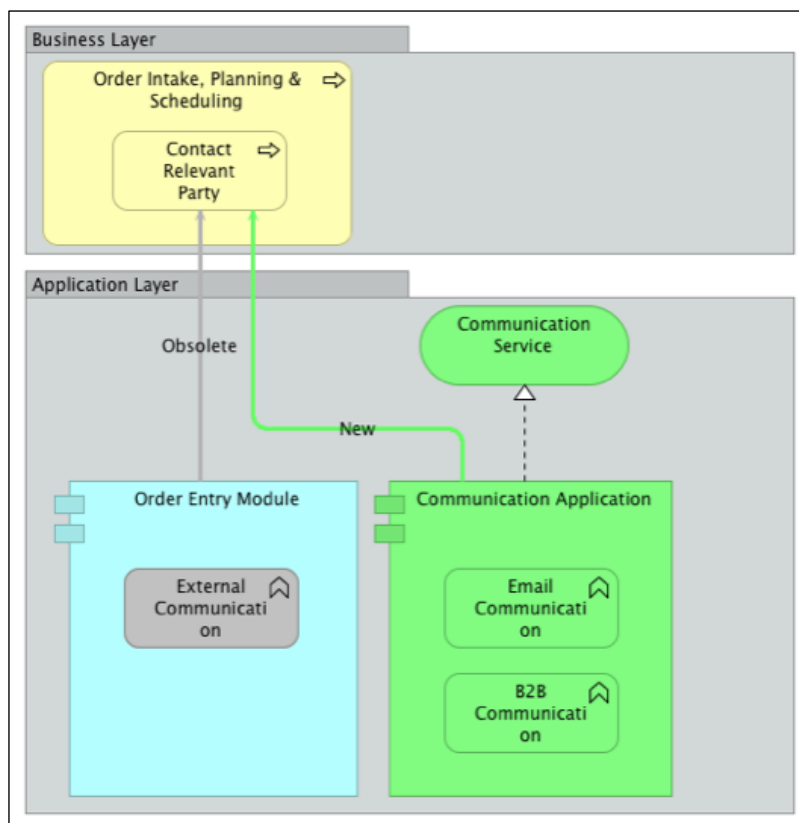


Figure 10: Example of Artifact 7: Gap of changes with focus on relations between business and application layers.

5 DISCUSSION

5.1 Visualization Principles

Our visualization principles are the minimum visual elements and the focus on changes. In order to follow these principles, we sought for means for separation the changes and unchanged elements and for visual-

ization of relations between unchanged elements and new elements at the business, application and technology layers. So, we explored available means of visualization in ArchiMate.

5.2 Means of Visualization

5.2.1 Changed Services. Using Colors and Labels <new>, <changed> and <obsolete>

First, we decided to use colors and show

- new objects in green;
- changed objects in orange;
- obsolete objects in grey.

However, we have found, that when a model contains a large number of objects and even objects over multiple layers, the use of colors can be distracting and can make the visualization unclear, due to the fact that the layers in ArchiMate are distinct by color. So, we suggest to include the labels <new>, <changed> and <obsolete> in the objects name. Obsolete objects also may be crossed, but this demands new extensions of ArchiMate.

5.2.2 Large Number of Changed Objects or Objects Related to the Changed Objects. Using Grouping, Aggregation and Composition

In the case of large amounts of changed elements and thus a large amount of colors, the elements belonging to a specific layer can be grouped together, indicating a specific layer, using the *group relation*.

Let us remind that “the grouping relationship is used to group an arbitrary group of model objects, which can be of the same type or of different types. In contrast to the aggregation or composition relationships, there is no “overall” object of which the grouped objects form a part. Unlike the other language concepts, grouping has no formal semantics. It is only used to show graphically that model elements have something in common. Model elements may belong to multiple (overlapping) groups.” (The Open Group, 2013)

Figure 8 demonstrates that the functions *E-mail Communication* and *B2B Communication* form the group *Communication Application*. At Figure 8 we also see the group *Order Entry* that includes the functions *Order Updating* and the obsolete function *External Communication*.

The application functions are visualised within the group *Application layer*; the motivation elements are presented within the *Motivation* group and the technological elements are situated in the group *Technology layer*.

Aggregation and composition in ArchiMate have the formal semantics inspired by the UML. When elements belong to other elements by an aggregation or composition relation, elements can be *rolled up to the*

aggregated or composed element. In our example, the requirements in Figure 3 can be *rolled up* to the top requirement *Requirements for Order Management*.

5.2.3 Omitting Unchanged Elements

The unchanged elements can be sometimes omitted from the model. When a model contains unchanged elements and those elements are part of a more generic element (that is required for clarity in the model), the elements can be left out and only the generic element can be a part of the model, using the generalization method.

Figure 10 shows the changes of the relations between the application and business layers: the relation between the function *External Communication* and the *Contact Relevant Party* is obsolete. The function *Order Updating* is omitted from the *Order Entry Module*, because the omitted function does not have any changed relations with business layer.

5.2.4 New Relations Between Changed Elements and Motivation

We have found it very useful that the core elements of ArchiMate can be related to motivational elements (goals, principles, requirements etc). (The Open Group, 2013) We associate the changed architectural elements with motivation elements to visualize the reasoning behind the changes.

Figure 8 shows that the motivating requirement *Order Entry must contain order updating functionality* is related to the *Order Updating* function. The principles *SaaS Unless Policy*, *Reuse before Buy before Build* and *Fit for Purpose and domain dedicated applications* motivate the *Online messaging platform vendor X (SaaS)*.

5.2.5 Relations <Extended-by>, <Replaced-by>.

The core of the transformation of ERP using the Best of Breed strategy is the decomposition of functionality and data, as it is shown in Figure 2. However, in reality, this pure transformation is rare. Usually, even when following the Best of Breed strategy of changes, some functions are deleted; the new functions replace the old ones, and the new functions are often added. Therefore, we have found it useful to introduce the relations between the new and old functions and groups of functions.

Relations like <Extended-by> and <Replaced-by> are definitely needed for the visualization of changes. Figure 7 shows such relationships. As the

current ArchiMate language does not have such relation types, we reuse the directed relations of types <used-by> or <triggering relationship> and replace their labels by <Extended-by>, <Replaced-by>.

5.3 Scalability of Visualization

In all cases, the visualization of changes is a very creative activity that demands good abstraction and generalization skills. The comprehensible views of a system with large amount of elements cannot be produced without abstractions. In particular, the ability to make groups of elements and the ability to responsibly omit elements are the key techniques for visualizing of changes in a comprehensible way.

Three types of abstractions have been found useful for visualization of changes:

1. the abstractions from the unchanged elements;
2. the abstraction from the elements that are out of focus of particular view;
3. the abstractions from relations and elements in a chain using the derived relationships of ArchiMate.

The abstractions from relations and elements in a chain using the derived relationships of ArchiMate need some explanation. The matter is that the structural relations in ArchiMate are divided into four categories of strength, where “association is the weakest structural relationship; composition is the strongest. Part of the language definition is an abstraction rule that states that two relationships that join at an intermediate element can be combined and replaced by the weaker of the two.”(The Open Group, 2013)

Using this abstraction rule of derived relations, a view may abstract from the intermediate elements of a chain of related elements (make some intermediate elements invisible). Elements can be rolled up, using the derived relationship of ArchiMate, where the chain of related elements can be generalized by relating two elements in the chain using the “weakest” relation in the chain.

For example, this derived relationship rule allows us to show in Figure 9 only the access relations between the services the *Order Entry Module* and the *Communication Application* and the messages (data objects) via the *Enterprise Service Bus*. All other technological components are omitted.

Another way of making the visualization of changes scalable is showing a sequence of gaps. As changes within a system are often implemented in steps, the visualization of each step as a gap of changes may restrict the number of changed elements

and relations and make the visualization of each gap comprehensible.

6 CONCLUSIONS AND FUTURE WORK

This paper presents a method for visualization of changes in ArchiMate with its motivation, migration and implementation extensions.

It was found that this combination of ArchiMate and extensions supports visualization of changes. We identified a set of views, and abstractions needed to visualize the changes in cases of transformation of ERP using the Best of Breed strategy. However, we also found a small set of new elements needed in ArchiMate to specify the relations between the obsolete and the new elements and to support abstraction from the obsolete and unchanged elements.

The reproducibility of the proposed visualization method was tested on several cases of transformation of ERP using the Best of Breed strategy. We expect that different strategy of changes may extend our method with new views on the gap of changes, but the core of the proposed method and discussed abstractions should remain the same.

In the future work, we are going to apply this method of visualization of changes in new projects. We plan to focus on the cases of application of derived relationships for visualization. Moreover, we plan to look at the methods for the separation of changes into sequential steps making the visualization of architectural changes scalable. Potentially, the proposed method may become a basis of a tool support for visualization of changes in ArchiMate.

REFERENCES

- Cardoso, J., Bostrom, R. P., Sheth, A., and Sheth, C. I. A. (2004). Workflow Management Systems and ERP Systems: Differences, Commonalities, and Applications. *Inf. Technol. Manage*, 5:319–338.
- Fritscher, B. and Pigneur, Y. (2011). *Advanced Information Systems Engineering Workshops: CAiSE 2011 International Workshops, London, UK, June 20-24, 2011. Proceedings*, chapter Business IT Alignment from Business Model to Enterprise Architecture, pages 4–15. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Institute of Educational Cybernetics (2012). *Archi 2.4*, <http://archi.cetis.ac.uk/>.
- Lankhorst, M. M., Proper, H. A., and Jonkers, H. (2009). *Enterprise, Business-Process and Information Systems Modeling: 10th International Workshop, BPMDS 2009, and 14th International Conference, EMM-*

SAD 2009, held at CAiSE 2009, Amsterdam, The Netherlands, June 8-9, 2009. Proceedings, chapter The Architecture of the ArchiMate Language, pages 367–380. Springer Berlin Heidelberg, Berlin, Heidelberg.

Lankhorst, M. M., Proper, H. A., and Jonkers, H. (2010). The Anatomy of the ArchiMate Language. *IJISMD*, 1(1):1–32.

Light, B., Holland, C. P., and Wills, K. (2001). ERP and Best of Breed: a comparative analysis. *Business Process Management Journal*, 7(3):216–224.

The Open Group (2013). *ArchiMate 2.1 Specification*. <http://pubs.opengroup.org/architecture/archimate2-doc/chap03.html>.

The Open Group (2016). TOGAF 9.1, an Open Group standard, <http://www.opengroup.org/subjectareas/enterprise/togaf>.

An in Depth Comparative Analysis of Software Tools for Modelling and Simulating Business Processes

Rallis C. Papademetriou¹ and Dimitrios A. Karras²

¹*Faculty of Technology, University of Portsmouth, Anglesea Road, Portsmouth, PO1 3DJ, U.K.*

²*Automation Department, Serea Hellas Institute of Technology, P.C. 34400 Psachna, Evoia, Greece
rallis.papademetriou@port.ac.uk, dakarras@teiste.gr*

Keywords: Business Process, Modelling Requirements, Comparative Analysis, Business Process Modelling Tools.

Abstract: Undoubtedly, Business process modelling is an increasingly popular research area for both organisations and enterprises due to its usefulness in facilitating better planning of resources, business reengineering and optimized business performance. The understanding of Business Process is an essential approach for an Organization to achieve set objectives and improve its operations. Business Process Modelling is a representation of the processes of the Organization. Several modelling techniques have been proposed and used to capture the characteristics of business processes. However, available techniques view business processes from different perspectives and have different features and capabilities. Furthermore, to date limited guidelines exist for selecting appropriate modelling techniques based on the characteristics of the problem and its requirements. Recent development has shown the importance of representing processes to carry out continuous improvement. The modelling and simulation of Business Processes has been able to show Business Analysts, and Managers where bottleneck exists in the system, how to optimize the Business Process to reduce cost of running the Organization, and the required resources needed for an Organization. This paper presents a comparative analysis of some popular business process modelling techniques. The comparative framework is based on the following major criteria: data flow capability, logical reasoning and understandability, specification of roles, and capability of simulation, flexibility and ease of use, simulation support and scope. The proposed framework can serve as the basis for evaluating further modelling techniques and generating selection procedures focusing on a comparison of existing tools to implement these techniques. This is to enable a potential user/modeller choose the right technique and tool in modelling critical Business Processes in order to analyse and optimize the operations of an Organization.

1 INTRODUCTION

Process is basically a part of everyday life. Process is involved in all actions carried out by both humans and organisation to achieve an objective. In organisations, value-added activities are becoming the central focus in the achievement of objectives set by an organisation. These value-added activities are referred to as Business process. Experts in Business analysis and Information System Analysis have both drawn the conclusion that the success of a system starts with a clear understanding and knowledge of the Business process of the organisation (Aguilar-Saven, 2003). The Business Process Model provides a platform for the analysis of the processes to be carried out. Business Process Modelling is a method commonly used in organizations to increase awareness of the business process, and to simplify the

complexity of the organization by disaggregating the process (Recker et al., 2009). The principles of Business Process Model can be traced back to the idea of Division of Labour in manufacturing by Adam Smith in 1776. According to Adam Smith, the breaking down of the process into specialised tasks, and the tasks assigned to different people in the chain of activities, would simplify and speed up the process. Although process modelling principles have been in existence since the 18th century, and the term Business Process Modelling was first used in the 1960s in an article, it only became popular in 1990s with the prevalence of Information Technology.

On its own, Business Process Modelling is not an improvement tool; it is simply a means to an end. It outlines the activities carried out in the organisation, providing a framework for improvement to be carried out. Business Process Modelling techniques are used

to design of the process model; and are implemented using software tools.

With the rapid growth and embrace in IT from the 1990s to this present time, BPM has gained more popularity in organisations. This has led to the growing number of Business Process Modelling tools, techniques, and methodologies. These different techniques of Business Process Modelling have their distinct uses and purposes. Due to the lack of guide, and a numerous presence of approaches, selection of the right tool and techniques has become more complex for organisations.

In view of this, a critical comparative analysis of the Business Process Modelling techniques and tools are carried out in this paper. This is to enable organizations to understand the right techniques and tools to use, as well as the pitfall of these techniques. Modelling of a Business process is used in detecting constraints of a system and providing a framework for carrying out continuous improvement of an Organization. This study aims at providing background knowledge of Business Process Management as an essential unit in carrying out continuous improvement, with focus on the Business Process Modelling Techniques and especially on the requirements and selection of proper software tools. Moreover, it seeks to determine and compare the modern mathematical and Information Systems based techniques and tools of Business Process modelling.

2 BUSINESS PROCESS MODELS COMPARATIVE ANALYSIS

A business model is a framework used to represent the complex reality of an organization. The business model is used to present a clear description of the objectives, strategies, organizational structure, and operations of an Organization. Business Models such as the Organizational Chart, and financial statements have been used by organizations for decades. In recent times, business models such as the Business Organizational Model, Business Rule Model, Business Motivational Models, and Business Process Model have been introduced (Bridgeland and Zahavi, 2009). These newer models do not displace the previous models, but to focus on other parts of the complex reality of a business.

The model of a business system can be classified into two categories (Kalnins et al.):

- i. As-is Model: This is a representation of the present state of the Business System which includes the Organizational Structure, the main

operations of the organization, the logical behaviour aspects of the system – who carries out a task, when it is to be carried out, and what it aims to achieved.

- ii. To-be Model: This is an improvement of the As-is Model after an analysis of the present state has been carried out. It is a representation of what the system can be and how the operations can be optimised.

A branch of Business Model which is considered to be an essential aspect of Business Process Management is the **Business Process Model**.

Process Model is used to represent a step-by-step approach in operations of an Organization. The Business Process model is a means to an end; showing what the series of activities aim to achieve, who/what department is to carry out the activities, and what stage the activity is to be carried out in the organizational operations process. The representation of the interactions between the tasks, operations, and processes provides a foundation for the improvement of the efficiency, effectiveness and business process of an organization (Business Process Modelling, 2007).

In present times, the research on Business Process Modelling is beginning to gain attention in both the academic and industrial environment. It is not just restricted to Business Process Management; it can be applicable to areas such as web development analysis, software development / engineering, and service-oriented structure (Prezel et al.). Business Process Modelling is not only used in conventional business environment, but also in government agencies, charity organizations, and all process-oriented systems. A Business Process Model can be used repeatedly for similar processes (Yamamoto et al., 2005). The increasing awareness and embrace of Information Technology/Information Systems has heightened the demand for process improvement with the aid of Business Process Modelling. The top priority of modelling a Business Process is to improve the system's performance through the optimisation of the activities carried out to achieve set objectives. Figure 1 illustrates the representation of the present activities of a system (AS-IS), and what the process is to become (TO-BE) order to optimize production/services. Other purposes to model a Business Process includes; communication, training, process reuse, persuasion in sales (Bridgeland and Zahavi, 2009).

The increase in demand of process modelling has equally made it as relevant as existing process optimization tool such as Total Quality Management (TQM). To model a Business Process, different

techniques can be used. These techniques have been developed for different purposes and are more suitable in various aspects. Business Process Modelling techniques are concerned with the mapping and workflow of activities in a system in order to be able to carry out analysis, and also provide a framework for change management (Business Process Modelling, 2013).

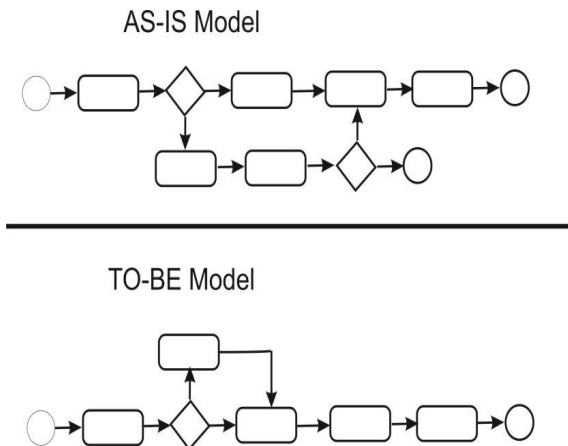


Figure 1: Example of an AS-IS and TO-BE Business Process Model.

The techniques of Business Process Modelling can be classified into three different set based on their representation of a model (Vergidis et al., 2008):

- *Diagrammatic Model*: At the inception of Business Process Model, these techniques were being solely used. It is a simple and easy-to-communicate graphical representation, originally developed for software analysis and development. Initially, the representation of these techniques was plain graphics.
- *Mathematical Model*: These are referred to as second generation of BPM techniques. These techniques are well defined and precise, and can be analysed mathematically to extract knowledge. Though these models are very appropriate in carrying out quantitative analysis and improvement of a process, they are not suitable in modelling complex constructs based on no emphasis on the diagrammatical representation of the process such as decision points. Representation of a Business process mathematically may prove to be much more complex than diagrammatic representation.
- *Business Process Languages*: The most recent set of BPM techniques to be developed. The development of these set of techniques, like the first generation set, was influenced by software

development. Business process languages is an Information Technology based technique in used to represent Business Processes. This latest trend of BPM techniques is dynamic in nature and rapidly evolving.

This section provides a comparative analysis of modern Business Process Modelling techniques. In this chapter, a basis for comparing the techniques is developed; the techniques are reviewed, stating their strengths and weakness (if any). The chapter ends with recent methodologies used in modelling an Enterprise.

In the study carried out, ten techniques were selected. This selection was based on popularity on websites of reputable Business Process Management software developers, and few existing recent journals from publishers such as IEEE, Science Direct.

In carrying out the comparative study of Business Process Model techniques, the answers to the following questions provided guidelines on how to achieve the objective:

- What are the bases for comparing the different Business Process Modelling techniques?
- What are the strengths and weakness in using the technique?
- What are the basic focuses of the techniques?

Past researchers have classified and compared the BPM techniques based on different views; in her article, Aguilar-Saven (2003) classified the techniques based on the purpose of the model and model change permissiveness (active and passive).

Another classification based on the representation of the techniques; Mathematical, Diagrammatical, and BP language (Vergidis et al., 2008).

However, most of these techniques make use of the conventional approach, representing Business Process (regardless of the form) as series of activities, emphasizing the structure of the flow of activities, and the resultant output from the activities; and also, used for similar purpose.

Hence, an approach for comparison is proposed based on the different techniques in this project to not only aid Academicians, but also aid BPM practitioners in selecting a technique.

In answering the research questions, four features were proposed; and the strengths and the weakness were stated. The features were based on intensive study of the general characteristics of Business Process Modelling techniques, and purpose of model classification by Aguilar-Saven (2003): *learning description; decision support for process design, control, and execution; Information Technology enactment support.*

The Table 1 below outlines and describes the features that were used in the comparison.

Table 1: Business Process Comparison Criteria.

Features	Description
Data Flow Approach	All techniques outline the flow of activities in the model; data flow description further explains what information is passed along. In some modelling techniques, text is combined with the visual representation of activities in the process flow. This approach is perhaps regarded as the most important form in the communication of process models to stakeholders. This feature explains the additional description offered by the techniques
Logical Approach	In this context, logical reasoning is the process of using statements/notations/arguments in describing conditions which must be met for an activity to be carried out. This feature adds more structure in the process model, providing alternate routes in achieving the objectives of the process. This feature is highly relevant in making decisions in the Organization
Role Approach	In process modelling techniques, this feature is usually a secondary structure included. The use of roles in techniques gives a clear description of who is responsible for the different tasks in the process, providing organization distribution of activities. The role assignment feature targets analysis of administrative procedures, providing a guideline in carrying out structural change in Organizations
Simulation	Simulation is the act of imitating the behaviour or operations of real-world Business Process or systems. This feature is used to carry out virtual analysis of a process

3 BUSINESS PROCESS MODELLING SOFTWARE TOOLS REQUIREMENTS AND COMPARATIVE ANALYSIS

Business Process Modelling/ Management software is the platform used in the implementation of the techniques discussed.

From researches on existing Business Process Modelling software available, BPM software are developed based on two distinct key functions:

1. Modelling of a Business Process.
2. Simulation of Business Process Model.
 - i. Analysis
 - ii. Optimization

Based on researches carried out in this work, it has been observed that some BPM software tools are just restricted to the design of the model, while some others have the capabilities of both functions. Some of the software applications have more than one technique embedded in them. In designing the Business Process Model, a modeller chooses the technique provided by the available software. For example, ARIS architecture discussed previously provides its own software which uses BPMN and EPC in the design of a Business Process Model. With the simulation function of the software, the results of the operations can be determined without necessarily spending huge capital of running the actual process in real-world. There are certain essential features which a Business Process Model should have:

- Complete Information: The resources, organizational units, idle time and operating time and everything related to the process should be stated.
- Realistic & Executable: The AS-IS model should be a real representation of the process, and also the TO-BE model should be achievable.
- Ability to be Partitioned: This is to enable analysis of a process unit to be carried out.
- Traceability: No activity should be without a connection to the start event.

Simulation mimics the actual process of a system. In Business Process context, simulation goes beyond imitating the system; simulation of a Business Process Model is carried out to achieve this given objectives in the Organization:

- > Process and resource analysis and optimization
- > Identification of Bottleneck
- > Human resources planning
- > Risk assessment and risk minimization.

The consideration whether Simulation of the Business Process is important in the organization is dependent of; the effort in preparing the process model and the aims to be achieved from simulation. The output of the simulation is a result of the model designed.

In carrying out the simulation, the following pieces of information are first determined:

- Processing time per function/ activity
- Number of available resources / equipments
- Work time for each of the resources / equipments
- Additional information in getting a precise result. E.g. interruptions due to pause in activities, parts replacement in the cause of the process.

The more detailed the information gathered, the more precise the simulation output.

In the search for Business Process software, at least fifty software applications were discovered. A question posed by Organization is “What is the best modelling software to choose?”

Based on researches carried out, below are different characteristics which are to be considered in choosing a business process modelling software:

- Notation and Technique
- Cost
- Operating System Capability
- Functionality
- Documentation format

Discussed below is a brief summary of some BPM software. The software discussed here is a mixture of both freeware and commercial version, therefore not all were tested. For this reason, some pieces of the information here are retrieved from the BPM software companies and third-party vendors:

1. **ADONIS:** This software was created and first released in 1995 by BOC Information Technologies Consulting, supporting Business Process Management based on a framework – BPMS which was developed at the University of Vienna. This software is free but a closed source which makes use of BPMN tool and with the commercial version providing simulation function. ADONIS software is created in an easy-to-use style so a beginner can easily understand how to use.
2. **Enterprise Architect Suite:** This software was developed by Sparx Systems based on UML. The Enterprise Architect was initially released in 2000 and supported just UML, but subsequent releases has seen it supporting basically all Object-Oriented technique – UML, BPMN, BPEL, SPEM. Aspects covered by this software not only include Business Process Modelling design, but also Simulation, Development Lifecycle, requirements management, project management. The software is a complete BPM suite which can be useful for software developers, modellers, Business Analyst, and other Organization entities.
3. **Microsoft Visio 2013:** The Microsoft Visio is more of a diagramming application which supports the design of a Business Process Model. This application first developed in 1992 by Shapeware Corporation, the product was later acquired by Microsoft. Microsoft Visio is not particularly based on a particular notation or technique. It provides different notation and techniques like IDEF0, BPMN2.0, UML; its own flowchart technique – Microsoft SharePoint

workflow; and different software and database diagrams. Using the Microsoft Visio, a modeller can decide to also design his notations.

4. **BIZAGI BPM Suite:** Bizagi BPM Suite was developed by Bizagi Ltd., a privately owned company founded in 1989. Bizagi BPM Suite consists of three tools – Process modeller, Studio, and Server. The Business Process are drawn and documented in the Process Modeller; the process applications are built in the studio and stored in a database, the execution and control of the built Process application takes place in the server. This all-in-one suite provides a graphical real-time tracking, simulation and monitoring of the process, providing Business Process management to companies such as Schlumberger, Adidas Group, Audi, and Petrobas. Bizagi supports BPMN, and also compatible with XML Definition Process Language (XPDL).

5. **Enterprise Dynamics:** The Enterprise dynamics software was developed by InControl Simulation Solutions. This application is a platform for carrying out simulation of a Business process. Models in this application are built by a drag-and-drop method. This software does not necessarily make use of any of the notations discussed, but it supports the model of Business Process through workflow technique. The main function of this software is to carry out simulation of a serial model. Though it is mainly used in the manufacturing sector, it can also be useful in other aspects of Industries. In the Enterprise Dynamics, the activities and roles are represented as servers. When simulated, the efficiency of the server (in %) is shown. Representing how the performance of each activity. A graph and simulation report is generated to carry out analysis.

This software makes use of a petri net - like and workflow technique in the representation of processes carried out, the blue circles represents the nodes in this process, moving from one server to another as the Business Process takes place.

This software is best suitable if a modeller is just concerned about the serial flow of activities and not concerned about the roles in carrying out the activities. Depending on the number of employees in carrying out tasks, the modelling of a process using this software can be very complex.

6. **Accuprocess Modeller:** The Accuprocess software is developed as an Easy-to-Use application compared to most other process modelling software. Accuprocess makes use of only one notation, BPMN 1.0 (similar to the flow

chart) which has few notations in carrying out a modelling exercise. This software provides free documents and trainings in which a new user can quickly learn how to use the software.

- ARIS Toolset:** ARIS toolset is made up of different products developed by IDS Scheer, a company established by Prof. August-Wilhelm Scheer, whose academic research brought about Architecture of Integrated Information Research (ARIS). IDS Scheer was developed to in 1984 to market the ARIS reference framework, but the company was later acquired by Software AG in 2010. The ARIS product was first released in 2009. Business Process Modelling on ARIS products is based on BPMN2.0 and EPC.

To further explain the techniques and tools of Business Process Modelling based on the comparative analysis, the modelling and simulation process in this subsection is carried out using software which makes use of a technique easily communicated and with simulation capability. Compared to other Business Process Software, Accuprocess Modeller provides full functionality (modelling & simulation) in its trial version

3.1 Case Study: Simulation Scenario

In this case scenario, only the activities directly involved with the receipt of goods of an organization are modelled. Ten actions are carried out with two decision points to determine the path to take. The scenario is outlined in table 3. In the decision points, probability of getting a defective batch is 20 %, good batch 80%; second decision point, probability of being damaged after being received (due to poor handling) is 10%, and being the fault of the supplier, 90%. The average time for the longest activity is 20 minutes, while the least activity takes 2 minutes. This process runs 600 times (i.e. 600 batches of goods are received).

Accuprocess Modeller. The Accuprocessor modeller is divided into two aspects; modeller and Simulation. In the modeller section, all the necessary notations to model the system are provided in a canvas. The modelling of the system is done by a drag-and-drop method.

A drag-and-drop canvas is provided, selecting the different objects to use in mapping the Business Process. Figure 2 depicts the process model of the Incoming goods received in the company. In the Accuprocess, the simulation is further sub-divided into two areas; the resource area, where the Organizational unit in which the roles can be assigned

to are situated – and the scenario- in which the simulation is to be carried out. Each activity is assigned to a role which is in the simulation resources. The vital activities to be part of the simulation exercise are included in the scenario, the execution time of each of these activities are further specified. Accuprocess gives the modeller option to specify the type of distribution, for purpose of the simulation carried out, a normal distribution is used.

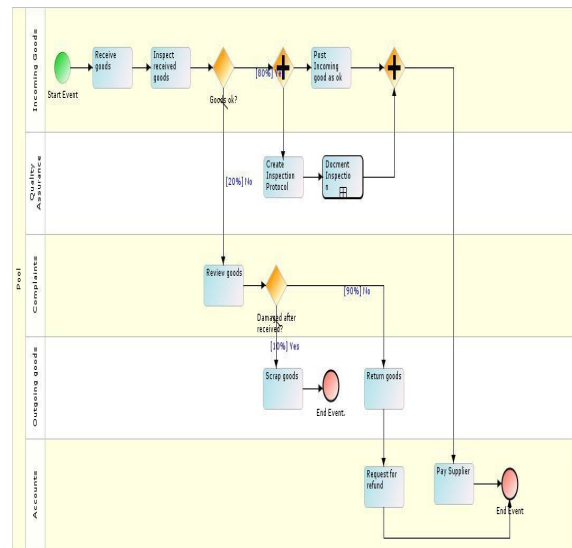


Figure 2: Incoming Goods process model in the relevant running scenario.

After the AS-IS configuration has been completed, a simulation is run to determine the state of operations in the Organization. A brief simulation run summary is displayed on the application interface. To view a detailed report to carry out analysis, an html file is generated displaying all the parameters and configurations. The complete steps in modelling and the simulation of the Business Process using the Accuprocess Modeller are included in the supporting document of this report. Figure 3 shows a brief summary of the simulation report.

Simulation Summary Report	
Scenario Name	Cost
Description	amount
ProcessFlow Name	Incoming goods
Resources Name	Goods_receipts
Comments	
Start Time	Fri Sep 05 12:31:00 BST 2014
End Time	Sun Sep 07 14:27:37 BST 2014
Total Cost	£5,225.69
Total Time Taken	49.94 Hours
No. of Runs Completed	600

Figure 3: Simulation Summary report.

Table 2: Business Process Modelling Software comparative characteristics evaluation.

	Notation/Techniques	Functionality	Operating System	Application Mode	Documentation Format
Adonis (Community Ed.)	BPMN 2.0	Model Design, Simulation	Windows	Stand-alone	Pdf, html
Enterprise Architect 11.1 Suite (Ultimate)	UML, SysML, BPMN 2.0, BPEL	Model Design, Simulation, Code Execution	Windows, Linux, Mac	Web, Stand-alone	Pdf, html, docx, rtf
Microsoft Visio 2013 (Professional)	BPMN 2.0, UML, IDEF0, Flowchart	Model Design, Simulation	Windows	Web, Stand-alone	vsdx, vsdm
Bizagi BPM Suite	BPMN 2.0	Model Design, Simulation, Code Execution.	Windows	Web	docx, xlsx
Enterprise Dynamics	Workflow, Petri net	Simulation	Windows	Stand- alone	Mod, bmp
ARIS Express	BPMN 2.0, EPC	Model Design	Windows, Linux, Mac	Stand- alone	Adf, pdf, rdf
ARIS Business Simulator	BPMN 2.0, EPC	Model Design, Simulation	Windows, Linux, Mac	Stand- alone, web	Adf, pdf, rdf
Accuprocess (Professional Ed.)	BPMN 1.0	Model Design, Simulation	Windows, Mac	Stand-Alone	Pm, Html, pdf, docx, jpg

Table 3: AS-IS Employee Configuration for the running scenario.

Organizational Units	Number of Employees	Cost per Employee £ (Hourly)	Activities Carried out	Duration Per Activity (minutes)
Incoming Goods	10	10	Receive goods	10
			Inspect goods	20
			Post goods as ok	2
Quality Assurance	4	15	Create Inspection Protocol	10
			Document protocol	excluded
Complaints	4	9	Review defective goods	20
Outgoing Goods Department	5	10	Scrap goods	5
			Return goods	5
Accounting	2	25	Refund Request	2
			Pay Supplier	2

Graphs generated from the simulation run are

shown in figure 4 depicting the result of the simulation carried out after 600 runs. The first graph shows the average utilization of the different units of the process in percentage, with Incoming Goods unit being more utilized. The run time graph depicts the maximum, minimum, and average time taken for one run to be completed in hours. The full summary report of the simulation can be viewed in the supporting documents attached to this report.

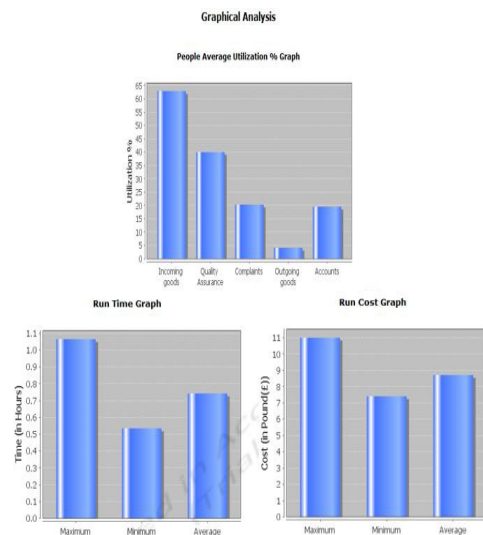


Figure 4: Graphical Analysis of the Model.

Advantages of Accuprocess Modeller.

- Easy to learn, and communicate the designed model
- The report generated is easily understandable
- Documentation of the model is achieved using a format *.jpg (for images), .docx (Microsoft word), html and pdf which is accessible in multiple platforms.
- It’s a stand-alone app; the models are easily designed without connecting to the internet. The designs are not stored in a database not operated by the Organization.
- It’s a light application easily downloaded and installed. Does not make use of an external database in its operations.
- Suitable for small and medium scale business

Disadvantages. In using the Accuprocess, only the hierarchical flow of activities can be modelled, not possible to model the structural hierarchy. This is due to the setback in using the BPMN, which is the only technique this software uses. Some software such as ARIS products, and Microsoft Visio, gives the modeller the Organization Chart option in the design.

In using a lesser suite or package, modelling and analysis is only carried out. But in carrying out the optimization of operations, physical observations have to be carried out, or by manually testing different values. In using the Enterprise edition, the optimization is automatically carried out by the software, saving the time and expenses carrying out the observations or changes would have cost. The disadvantages stated here is not restricted to Accuprocess Professional Edition, but some other less expensive Business Process Software.

ARIS Business Simulator. A similar process is being modelled and simulated using ARIS Business Simulator. This is done to show the ability to optimize process automatically. The optimization aim is to determine the least number of employees to carry out each run of the process at a faster time with less cost. It should be noted that the simulation exercise using the ARIS Business Simulator was not carried by the author. It was obtained from the ARIS forum to carry out the comparison. Therefore some parameters and configurations will have some slight changes, but still similar approaches.

Based on the configuration of the process, the cost and time in operations vary. During the process analysis, the effect of different process configuration and resources are checked in a simulation environment. These configurations are automatically created by the simulator. The results obtained are saved to be compared and evaluated for the optimization of the process.

The type of Business Process Model technique is specified, in this case EPC is used; the database to use is selected; and the model in which to carry out the simulation is defined. In carrying out the experiment, the attribute of the objects is varied, the highest and lowest limits of the objects (in this case, all the Organizational Units) are specified; the low and high text field. This is referred to as Factor Variation.

Depending on the configuration in the Factor Variation, many scenarios are created.

Table 4: TO-BE Employee Optimization Configuration.

Organizational Units	Minimum Staff	Maximum Staff
Incoming Goods Department	1	15
Accounting	1	10
Quality Assurance	1	10
Complaints Department	1	10
Outgoing Goods Department	1	10

The next step is to specify the responses to be saved, in this case scenario, the throughput time for the model (Receipt of Goods), and the idle time cost

for the objects (the Organizational units).

In carrying out the simulation, the optimization configuration has to be specified according to what is to be achieved. In this case, minimization of the model and objects is specified.

In the ARIS software, one of the documentation format used is Microsoft Excel; this makes the analysis to be carried out easier. After all parameters has been set in the simulation the output format-Excel- is specified, and a file is created for the result to be viewed and saved.

The generated output Microsoft Excel file of the simulation is displayed, showing the values for the different objects, the factors, and responses based on the amount of possible configuration being run.

The optimization configuration is compared from the values in the objective column which is generated automatically. These values are further compared to idle time cost of the different responses. This process can be made easier by sorting the values according to ascending or descending order, or by generating a graph based on the values.

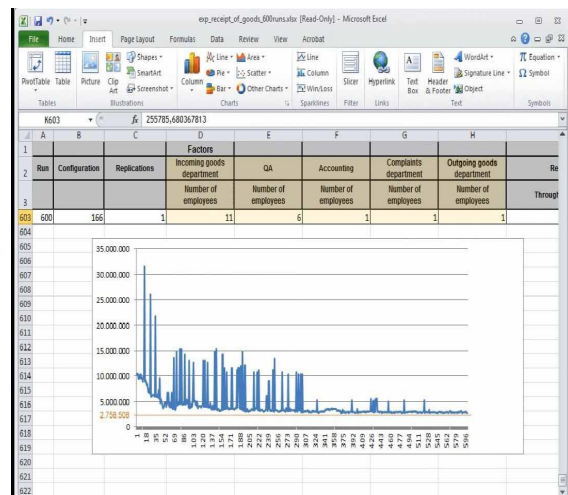


Figure 5: ARIS software provided Optimization graph.

The lowest objective value which is the idle time cost provides the result for the factors. From figure 5 above the result was gotten in the configuration run 166 with the factors values as;

Table 5: Required numbers of employees.

Organizational Unit	Employees Required
Incoming Goods Department	11
Quality Assurance	6
Accounts	1
Complaints Department	1
Outgoing Goods Department	1

Table 5 shows the required staff of the different units needed to receive goods at the lowest possible throughput time at the least idle time cost.

Advantages.

- Suitable for the large scale businesses, and for modelling the Enterprise
- Used in illustrating the structural hierarchy of the enterprise
- Gives the user ability to automatically optimize the process

Disadvantage.

- A major disadvantage in using the EPC which is the major techniques used in this software is the specification of roles for every function/carried out by adding the Organization unit notation. Using this makes the model quite messy compared to the arranged format in using the BPMN.

4 DISCUSSION - CONCLUSIONS

Business Process Model as a topic aims at improving the operations of an Organization. However, the presence of numerous techniques and tools has created a setback to modelling a Business Process. In this study, a comparison have been carried out to enable a potential user understand the usability of these techniques and tools.

The techniques were compared using features necessary for having a complete Business Process Model; *data flow, logical reasoning, specification of roles, and capability of simulation*. A technique possessing all the features will be effective for use in these four target areas of an Organization:

- Production Planning: Goods/services to be produced – the conversion of the input into output.
- Organizational Structure: With the structural view, the management determines the relationship between the tasks and equipment/personnel/units; useful for the Human resource unit of an Organization.
- Logistics: The flow of products; not only how it’s being produced in the Organization, but also to the point of consumption, how the requirements are met.
- Strategic Planning: The strategic planning involves a futuristic plan for the Organization. Using the techniques and tools, the AS-IS model is analysed, improved upon to generate the TO-BE model. Simulation is carried out to determine the result of the future plan. This is done to reduce

cost of actually implementing a plan that might be a failure.

Though techniques such as EPC, IDEF, WORKFLOW, and BPMN all have the features mentioned, it is necessary for a modeller to consider one which is easily understandable to stakeholders; easy in the communication of processes involved in the business, and to also facilitate learning. From the study, the EPC and the BPMN technique clearly states the activities, roles, the flow of messages (in BPMN), documents to be read, and requirements for a task to be carried out. These two techniques also give a clear outline of the structure of an Organization.

Some major software tools were analysed in this study to determine what requirements to look out for in choosing Business Process software. The comparison showed that most software packages make use of BPMN technique in the modelling of a Business Process. Judging from the outcome of this study, it is best to choose a software tool which not only models a system, but, also, it is able to carry out its simulation. It was shown that some software tools with the simulation functionality do provide only analysis of the AS-IS model, while others go further to enable optimization of the Business Process to be carried out producing the TO-BE Business Process Model.

This study ended with simple simulation scenarios carried out to show the importance of modelling and simulating a Business Process to improve the operations of an Organization.

Undoubtedly, Business Process Modelling is relevant in improving a system, and researches carried out in relation to this topic are still very active. Nevertheless, some important areas in relation to this topic are yet to be covered, as:

- BPM techniques are most times mixed with other modelling techniques such as Software and Data Modelling, though some techniques are applicable to different process model. This makes it difficult to distinguish which is most suitable for Business Process.
- Standardization of Business Process Model software due to the presence of numerous tools.
- With the popularity of Lean Six Sigma, a future research on the Integration of Business Process Management and Lean Six Sigma is to be carried out to further strengthen the techniques used in carrying out continuous improvement of an Organization.

REFERENCES

- Aguilar-Saven, R. S. (2003). *Business Process Modelling: Review and Framework*. International Journal of Production Economics (90), 129-149.
- Badica, C., Badica, A., & Litoiu, V. (2003). Role Activity Diagrams as Finite State Process. *Second International Symposium on Parallel and Distributed Computing (pp. 15-22)*. IEEE.
- Bridgeland, D. M., & Zahavi, R. (2009). *Business Modelling - A Practical Guide to Realizing Business Value*. Burlington: Morgan Kaufmann.
- Business Process Modelling. (2007). Retrieved 06 21, 2014, from Business Process Modelling: <http://www.businessprocessmodelling.co.uk>
- Business Process Modelling. (2013). Retrieved June 24, 2014, from BusinessBalls.com: <http://www.businessballs.com/business-process-modelling.htm>
- Kalnins, A., Kalnina, D., & Kalis, A. (n.d.). Comparison of Tools and Languages for Business Process Reengineering. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.127.1036&rep=rep1&type=pdf>
- Prezel, V., Gašević, D., & Milanović, M. (n.d.). Representational Analysis of Business Process and Business Rule Languages. Retrieved 06 04, 2014, from OntoRule Project: http://ontorule-project.eu/attachments/075_buro2010_paper_2.pdf
- Recker, J., Indulska, M., Rosemann, M., & Green, P. (2009). *Business Process Modelling - A Comparative Analysis*. Journal of the Association for Information System, 10 (4), 333-363.
- Vergidis, K., Tiwari, A., & Majeed, B. (2008). *Business Process Analysis and Optimization: Beyond Reengineering*. IEEE Transactions on systems, man, and cybernetics—part c: applications and reviews, 69-82.
- Yamamoto, R., Yamamoto, K., Ohashi, K., & Inomata, J. (2005). Development of a Business Process Modeling Methodology and a Tool for sharing Business Processes. *12th Asia-Pacific Software Engineering Conference (APSEC'05)*, IEEE.

Microflows: Lightweight Automated Planning and Enactment of Workflows Comprising Semantically-Annotated Microservices

Roy Oberhauser

*Computer Science Department, Aalen University, Aalen, Germany
roy.oberhauser@hs-aalen.de*

Keywords: Workflow Management Systems, Microservices, Service Orchestration, Agent Systems, Semantic Technology.

Abstract: Business processes are facing increasing pressure to quickly and flexibly adapt to changes in the process context. Moreover, microservices are becoming increasingly popular as an architectural style for partitioning business logic into small services accessible with lightweight mechanisms, leading to increasing pressure for a more dynamic integration of information services with processes. Process-aware information systems must thus increasingly incorporate the ability to react to unforeseen changes during process enactment, facing difficulties in pre-modelling all the possible process variations and enactment circumstances for larger process models. This paper presents Microflows, an automatic lightweight declarative approach for the workflow-centric orchestration of semantically-annotated microservices using agent-based clients, graph-based methods, and the lightweight semantic vocabularies JSON-LD and Hydra. The evaluation results show the approach's potential in lightweight resource utilization, investigates its scalability, and compares its automation to common manual workflow modeling and enactment.

1 INTRODUCTION

In many areas of society today, a trend towards increased automation can be observed. One area in particular is that known as business processes or workflows. As one indicator of its importance to business, spending on Business Process Management Systems (BPMS) is forecast at \$2.7 billion in 2015 (Gartner, 2015). The automation of a business process according to a set of procedural rules is known as a workflow (WfMC, 1999). In turn, a workflow management system (WfMS) defines, creates, and manages the execution of workflows (WfMC, 1999). These workflows are often rigid, and while adaptive WfMS can handle certain adaptations, they usually involve manually intervention to determine the appropriate adaptation.

Moreover, there is an increasing trend toward applying the microservice architecture style (Fowler, and Lewis, 2014) for an agile and loosely coupled partitioning of business logic into small services accessible with lightweight mechanisms. They can be deployed independently of each other and conform to a bounded context. As the dynamicity of the service world grows, the need for more

automated and dynamic approaches to service orchestration becomes evident.

Service orchestration represents a single executable process that uses a flow description (such as WS-BPEL) to coordinate service interaction orchestrated from a single endpoint, whereas service choreography involves a decentralized collaborative interaction of services (Bouguettaya et al., 2014), while service composition involves the static or dynamic aggregation and binding of services into some abstract composite process.

While automated and dynamic workflow planning can remove the manual overhead for workflow modeling, a fully automated semantic integration process remains challenging, with one study indicates that it is achieved by only 11% of Semantic Web applications (Heitmann et al., 2012). Rather than pursuing the fairly heavyweight service-oriented architecture (SOA) and semantic web standards, we chose to investigate the viability of a lightweight approach. Analogous to microservices principles, we use the term microflow to mean lightweight workflow planning and enactment of microservices, i.e. a lightweight service orchestration of microservices.

This paper explores an approach we call Microflows for automatically planning and enacting lightweight dynamic workflows of semantically annotated microservices. It uses a declarative paradigm with cognitive agents leveraging current lightweight semantic and microservice technology and investigates its viability. Note that this approach does not intend to address all facets of BPMS support, but is focused on a narrow area addressing the automatic orchestration of dynamic workflows given a multitude of microservices using a pragmatic lightweight approach rather than a theoretical treatise.

This paper is organized as follows: the next section discusses related work. Section 3 and 4 describe the solution approach and its realization respectively. Section 5 describes the evaluation, followed by the conclusion.

2 RELATED WORK

While the term microflow has been used in IBM business process manager documentation to mean a transient non-interruptible BPEL process (IBM, 2015), in our terminology a microflow is independent of any specific BPMS or any choreography or orchestration language.

Work related to the orchestration of microservices includes (Rajasekar et al., 2012), who describe the integrated Rule Oriented Data System (iRODS) for large-scale data management, which uses a distributed event-condition-action rule engine to orchestrate micro-services into conditional chain-oriented workflows, maintaining transactional properties through recovery micro-services. (Alpers et al., 2015) describe a microservice architecture for BPM tools, highlighting a Petri Net editor to support humans with BPM.

As to web service composition, (Sheng et al., 2014) provides a survey of current research prototypes and standards in the area of web service composition. While the web service composition using the workflow technique (Rao and Su, 2004) can be viewed having similarity to ours, our approach does not explicitly create an abstract composite service but rather can be viewed as automated dynamic web service orchestration using the workflow technique.

Concerning the combination of multi-agent systems and microservices, (Florio, 2015) proposes a multi-agent system for decentralized self-adaptation of autonomous distributed components (Docker-based microservices) to address scalability,

fault tolerance, and resource consumption. These agents known as selfLets mediate service decisions using partial knowledge and exchanging messages. (Toffetti et al., 2015) provide a position paper focusing on microservice monitoring and proposing an architecture for scalable and resilient self-management of microservices by integrating management functions into the microservices, wherein service orchestration is cited to be an abstraction of deployment automation (Karagiannis et al., 2014), microservice composition or orchestration are not addressed.

Related standards include OWL-S (Semantic Markup for Web Services), an ontology of services for automatic web service discovery, invocation, and composition (Martin et al., 2004). Combining semantic technology with microservices, (Anderson et al., 2015) present an OWL-centric framework to create context-aware applications, integrating microservices to aggregate and process context information. For a more lightweight semantic description of microservices, JSON-LD (Lanthaler and Gütl, 2012) and Hydra (Lanthaler, 2013) (Lanthaler and Gütl, 2013) provide a lightweight vocabulary for hypermedia-driven Web APIs and enable the creation of generic API clients.

In contrast to the above work, our contribution specifically focuses on microservices, proposing and investigating an automatic lightweight declarative approach for the workflow-centric orchestration of microservices using agent-based clients, graph-based methods, and lightweight semantic vocabularies like JSON-LD and Hydra.

3 SOLUTION APPROACH

The principles and process constituting the solution approach described below reference the solution architecture of Figure 1.

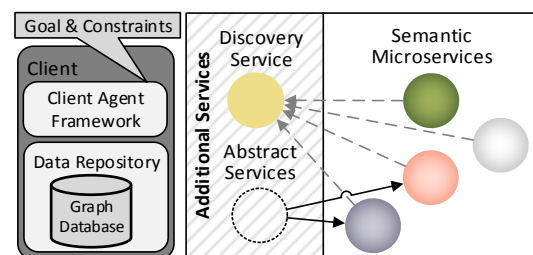


Figure 1: Solution concept.

3.1 Microflow Solution Principles

The solution approach consists of the following

principles:

Semantic Self-description Principle: microservices provide sufficient semantic metadata to support autonomous client invocation. For example, in our realization this was done using by using JSON-LD with Hydra.

Client Agent Principle: Intelligent agents exhibit reactivity, proactiveness, and social ability, managing a model of their environment and can plan their actions and undertake goal-oriented behavior (Wooldridge, 2009). Nominal WfMS are typically passive, executing a workflow according to a manually determined plan (workflow schema). Because of the expected scale in the number of possible microservices, the required goal-oriented choices in workflow modeling and planning, and the autonomous goal-directed action required during enactment, agent technology seems appropriate. Specifically, we chose Belief-Desire-Intention (BDI) agents (Bratman et al., 1988) for the client realization, providing belief (knowledge), desire via goals, and intention utilizing generated plans that are the workflow.

Graph of Microservices Principle: microservices are mapped to nodes in a graph and can be stored in a graph database. Nodes in the graph are used to represent any workflow activity, such as a microservice. Nodes are annotated with properties. Directed edges depict the directed connections (flows) between activities annotated via properties. To reduce redundant resource usage via multiple database instances, the graph database could be shared by the clients as an additional microservice.

Microflow as Graph Path Principle: A directed graph of nodes corresponds to a workflow, a sequence of operations on those microservices, and is determined by an algorithm applied to the graph, such as shortest path. The enactment of the workflow involves the invocation of microservices, with inputs and outputs retained in the client and corresponding to the client state.

Declarative Principle: any workflow requirement specifications take the form of declarative statements, such as the starting microservice type, end microservice type, and constraints such as sequencing constraints.

Microservice Discovery Service Principle (Optional): we assume a microservice landscape to be much more dynamic in microservices coming and going than heavyweight services, and therefore utilize a microservice registry and discovery service. This could be deployed in different ways, including centralized, distributed, or having it embedded

within each client, and utilize voluntary microservice-triggered registration or multicast mechanisms. For security purposes, there may be a wish to avoid discovery (of undocumented microservices) and thus maintain a whitelist. Clients may or may not have a priori knowledge of a particular microservice. Various broadcast services could be used.

Abstract Microservices Principle (Optional): microservices with similar functionality (search, hotel booking, flight booking, etc.) can be grouped behind an abstract microservice. This provides an optional level of hierarchy to allow concrete microservices to only provide a client with link to the next abstract microservice(s), since the actual concrete microservices can be numerous and rapidly change, while determining exactly which one is appropriate can best be done by the client in conjunction with the abstract microservice.

Note that the Data Repository and Graph Database could readily be shared as a common service, and need not be confined to the Client

3.2 Microflow Lifecycle

The microflow lifecycle involves three stages as shown in Figure.



Figure 2: Microflow lifecycle.

The *Microservice Discovery* stage involves utilizing a microservice discovery service to build a graph of nodes containing the properties of the microservices and links to other microservices. This is analogous to mapping the landscape.

In the *Microflow Planning* stage, an agent takes the goal and other constraints and creates a plan known as a microflow, finding an appropriate start and end node and using an algorithm such as shortest path to determine a directed path.

In our opinion, a completely dynamic enactment without any planning (no schema) could readily lead to dead-end paths causing a waste of unnecessary invocations that do not lead to the desired goal and can potentially not be undone. This is analogous to following hyperlinks without a plan, which do not lead to the goal and require backtracking. Alternatively, replanning after each microservice invocation involves planning resource overhead (CPU, memory, network), and since this is unlikely to dynamically change between the start and end of this lifecycle, we chose the pragmatic and hopefully

more lightweight approach from the resource utilization perspective: plan once and then enact until an exception occurs, at which point a necessary replanning is triggered. Further advantages of our approach in contrast to a thoroughly adhoc approach is that the client is assured that there is at least one path to the goal, and validation of various structural, semantic, and syntactic aspects can be readily performed.

In the *Microflow Enactment* stage, the microflow is executed by invoking each microservice in the order of the plan, typically sequentially but it could involve parallel invocations. A replanning of the remaining microflow can be performed if an exception occurs or if notified by the discovery service of changes to the set of microservices. A client should retain the microflow model (plan) and be able to utilize the service interfaces and thus have sufficient semantic knowledge for enactment.

The *Microflow Analysis* stage involves the monitoring, analysis, and mining of execution logs in order to improve future planning. This could be local, in a trusted environment, or this could be distributed. Thus, if invocation of a microservice has often resulted in exceptions, future planning for this client or other clients could avoid this troublesome microservice. Furthermore, the actual latency incurred for usage of a microservice could be tracked and shared between agents and taken into account as a type of cost in the graph algorithm.

4 REALIZATION

A realization of the solution concept as a prototype involved mapping technology choices onto the solution concept (Figure) and explained below.

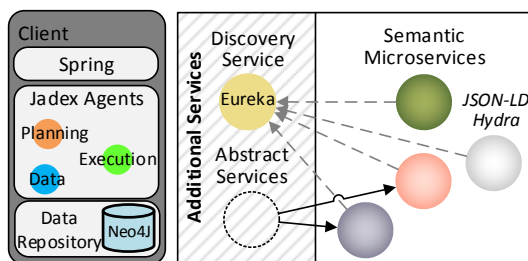


Figure 3: Microflow solution realization technologies.

The prototype integrates the following, especially for REST (REpresentational State Transfer) and HATEOAS support (Fielding, 2000): Spring-boot-starter-web v. 1.2.4, which includes Spring boot 1.2.4, Spring-core and Spring-web v.

4.1.6, Embedded Tomcat v. 8.0.23; Hydra-spring v. 0.2.0-beta3; and Spring-hateoas v. 0.16. For JSON (de)serialization Gson v. 2.6.1 is used. Unirest v. 1.3.0 is used to send HTTP requests.

4.1 Microservices

A code snippet of the Spring-based controller for the microservices is shown in Figure 4. Followers was explicitly provided to avoid having to know how to invoke domain-specific microservice operations when only the potential followers are of interest.

```

1 @Controller
2 public class PreferencesMicroServiceController {
3     @Autowired
4     private HttpServletRequest context;
5     @RequestMapping(value = "/", method = RequestMethod.GET)
6     public @ResponseBody Resource<Service> getDescriptionFromBasePath() {
7         return getDescription();
8     }
9     @RequestMapping(value = "/description", method = RequestMethod.GET)
10    public @ResponseBody Resource<Service> getDescription() {
11        ...
12    }
13    @RequestMapping(value = "/followers", method = RequestMethod.GET)
14    public @ResponseBody Resource<ServiceFollowers> getFollowers() {
15        ...
16    }
17    @RequestMapping(value = "/execute", method = RequestMethod.PUT)
18    public @ResponseBody Resource<ItemList> executePreference(@RequestBody
19        URL url) {
20        ...
21    }
22 }

```

Figure 4: Example microservice Spring controller.

An example microservice description using JSON-LD and Hydra is shown in Figure 5.

```

{
  "@context": {
    "@vocab": "http://schema.org/"
  },
  "@type": "Service",
  "serviceType": "Preferences",
  "serviceOutput": {
    "@type": "ItemList",
    "itemListElement": {
      "@type": "PropertyValue",
      "unitText": "",
      "value": "",
      "name": ""
    },
    "numberOfItems": 0
  },
  "offers": {
    "@type": "Offer",
    "businessFunction": "http://purl.org/goodrelations/v1#ProvideService"
  },
  "availableChannel": {
    "@type": "ServiceChannel",
    "serviceUrl": "http://192.168.0.14:8333/execute"
  },
  "description": "Get preferences service",
  "id": "http://192.168.0.14:8333/",
  "hydra:followers": {
    "id": "http://localhost:8333/followers",
    "hydra:operation": [
      {
        "hydra:method": "GET"
      }
    ]
  },
  "hydra:execute": {
    "id": "http://localhost:8333/execute",
    "hydra:operation": [
      {
        "hydra:method": "PUT",
        "hydra:expects": {
          "@type": "URL",
          "hydra:supportedProperty": []
        }
      }
    ]
  }
}

```

Figure 5: Example microservice description with Hydra.

To support a larger-scale evaluation of the prototype, we created virtual microservices that differentiate themselves semantically but provide no real functionality. As a REST-based discovery service, Netflix’s open source Eureka (Eureka, 2016) v. 1.1.147 is used.

4.2 Microservice Client

For the client, Jadex v. 3.0-SNAPSHOT is used as a BDI agent framework (Pokahr, Braubach, & Lamersdorf, 2005). Jadex’s BDI nomenclature consists of Goals (Desires), Plans (Intentions), and Beliefs. Beliefs can be represented by attributes like lists and maps. Three agents were created: the DataAgent is responsible for providing for and maintaining data repository, the PlanningAgent generates a path through the graph as a microflow, while the ExecutionAgent communicates directly with microservices to invoke them according to the microflow. For the client’s Data Repository, Neo4j and Neo4j-Server v. 2.3.2 is used.

4.3 Microflow Lifecycle

The goals and constraints are referred to as PathParameters and consist of the startServiceType (e.g., preferences), endServiceType (e.g., payment), and constraint tuples in JSON as shown in Figure 6. Each constraint tuple consists of the target of the constraint (the service type affected), the constraint, and a constraint type (required, beforeNode, afterNode). For instance, target = "Book Hotel", constraint = "Search Hotel", and constraint type = "afterNode" would be read as: "BookHotel" after "Search Hotel", implying the microflow sequencing must ensure that "Search Hotel" precedes "Book Hotel" (but must not be directly before it).

```
{
  "startServiceType": "Preferences",
  "endServiceType": "Payment",
  "constraints": [
    {
      "type": "RequiredNode", "target": "Flight Search",
    },
    {
      "type": "AfterNode", "target": "Payment", "constraint": "Book Hotel",
    },
    {
      "type": "BeforeNode", "target": "Hotel Search", "constraint": "Book Hotel",
    },
    {
      "type": "AfterNode", "target": "Payment", "constraint": "Book Flight"
    }
  ]
}
```

Figure 6: Goal and constraints inputs in JSON.

These set of constraint tuples are analyzed, whereby any AfterNode is converted to a BeforeNode by swapping target and constraint, then ordered, and then checked if any constraint is redundant. Then RequiredNode constraints are also converted to BeforeNode constraints.

We used a PathWrapper because of occasional issues incurred when passing Path objects in the Neo4J format between agents.

4.3.1 Microservice Discovery Stage

The *Microservice Discovery* stage involves the interactions shown in Figure 7, where Microservices first register themselves with the DiscoveryService. On client initialization, the DataAgent has the DataRepository fetch (via its DatabaseController) the registered services from the DiscoveryService and retrieve the service description from each microservice rather than a central repository. This avoids the issues of the discovery service retaining duplicate or incorrect (stale) semantic data.

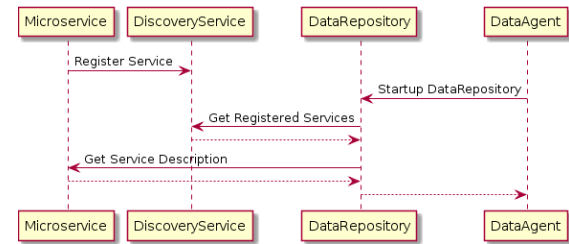


Figure 7: Microservice Discovery stage interactions.

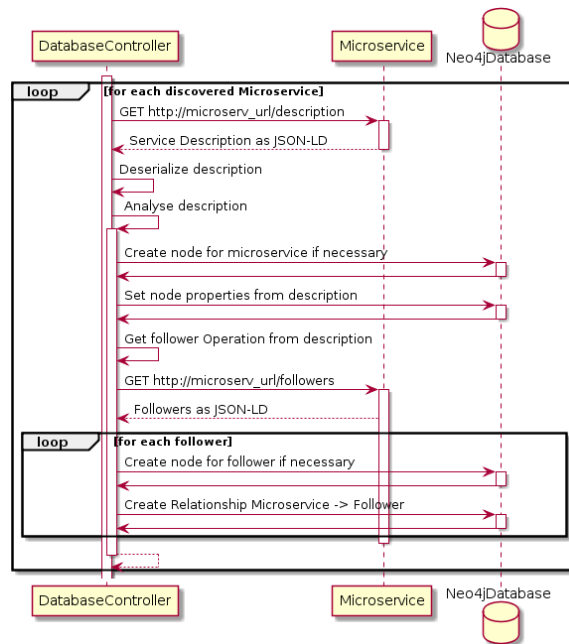


Figure 8: Microservice description collection interactions.

In Figure 8, the semantic description of the microservice is retrieved and, if a node does not yet exist, a node is inserted in the graph along with its properties. All followers are also inserted (if not already) and their association with this microservice is annotated as a directed edge. If any microservices are detected that were not (yet) registered with the discovery service, these are also tracked in a list.

4.3.2 Microservice Planning Stage

During the *Microservice Planning* stage, the *PlanningAgent* plans a microflow. It has two Beliefs: *PathParameters* (the input) and the *Path*. The annotations show that anytime *PathParameters* changes, *Jadex* triggers a planning.

```

1 @Plan(trigger = @Trigger(factchangeds = "pathParameters"))
2 public void pathParametersChanged(ChangeEvent event) {
3   if (algorithmService != null) {
4     validPaths = algorithmService.getShortestPaths(param);

```

Figure 9: Microflow planning triggering.

Although *Neo4J* offered native graph algorithms, they did not completely fulfill our requirements. While we utilize them, we generate microflows with our own algorithm as shown in Figure 10. After converting the constraints (Line 1-3) as described above, the set of possible starting microservices matching the starting type are determined (Line 4). Then this set is iterated over using the *shortestPath* algorithm, trying to find a path to the start of the next *pathPart*, which is either the target of the next constraint or the *endServiceType*, which is iterated (Line 7) since multiple nodes are possible. Then a recursive calculation of *pathParts* is initiated (Line 10), which either ends due to a deadend (Line 17) or the path to a valid *endServiceType* being found (Line 15).

```

1 constraints = analyze(constraints);
2 // list of constraints
3 cList = orderConstraints(constraints);
4 startList = findServicesForServiceType(startType);
5 foreach(service in startList) {
6   nextTargetList = findServicesForServiceType(cList[0]);
7   for(target in nextTargetList){
8     path = findPath(service, target);
9     if(path.isValid()){
10      pathPartList = calculateNextPathPart(target,
11      cList[1..cList.length()-1]);
12      if(pathPartList.isValid()){
13        pathPartList.prepend(path);
14        possiblePathsList.add(pathPartsToPath(pathPartList));
15        break; // Stop, valid path from a start found
16      }
17    }
18  }
19 }
20 calculatedPath = findBestPath(possiblePathsList);
21 return calculatedPath;

```

Figure 10: Microflow generation algorithm (pseudocode).

The microflow schema is currently only applicable for the current enactment, so that future enactments involve a replanning. However, the microflow schema (sequence plans) could be retained and reused if desired - for instance, if nothing changed in the environment. If multiple clients and thus agents coexisted in a trusted environment, then they could utilize their social communication ability to request and share

microflows.

Although support for gateways (forking and merging) and intermediate events are feasible in this approach, are prototype did not yet realize this functionality at this time. Support for using costs with graph paths is implemented but not utilized in our evaluation, since with virtual microservices it appeared artificial for the focus of our investigation.

In focusing on a lightweight approach, and not requiring interoperability, we chose to avoid the XML-centric *BPEL* and *BPMN*, which would only have added extra overhead in our case study without any benefit.

4.3.3 Microservice Enactment Stage

For the *Microflow Enactment* stage, the *ExecutionAgent* is primarily responsible. It has three beliefs: *pathWrapper*, *currentNode* (points to which node is either active or about to be executed), and *path* (the planned microflow), and similar to Figure 9, the *ExecutionAgent*'s plan is triggered by a change to the path variable (by the *PlanningAgent*), as shown in Figure 11.

```

1 @AgentArgument
2 @Belief
3 protected PathWrapper pathwrapper;
4 @Belief
5 protected int currentNode = -1;
6 @Belief
7 protected Path path;
8
9 @Plan(trigger = @Trigger(factchangeds = "path") )
10 public void startPathExecution(ChangeEvent event) {

```

Figure 11: ExecutionAgent (snippet).

The microflow enactment algorithm is shown in Figure 12. Line 8 shows that abstract nodes are skipped. Line 14 is a loop for the case when a microservice takes more than one input. In Line 17 the output of this invocation is retained for possible input as client state during further microflow processing. Because the microservice invocations are asynchronous, a *Java CountdownLatch* is used for synchronization purposes. Line 19 shows that a new microflow planning starting with the current node is triggered when an error occurs with avoidance of the problematic microservice if possible (e.g., if other identical microservice types are available) - otherwise a retry can be attempted. In addition, the initial constraints are readjusted since certain constraints may no longer be applicable (e.g., if they were already fulfilled in the partial microflow already executed).

```

1 // toBeExecutedNode = current node index in workflow/path
2 // possibleInputList = available inputs from previous services
3
4 if(!isNodeValid(toBeExecutedNode))
5     return;
6 serviceDescription = getServiceDescription(toBeExecutedNode);
7 if(isNodeAbstract(serviceDescription)){
8     toBeExecutedNode++; // continue with next node in workflow
9     return;
10 }
11 if(!isValidInputAvailable(serviceDescription))
12     return;
13
14 foreach(input in getValidInputList(serviceDescription)){
15     response= executeServiceWithInput(serviceDescription, input);
16     if (response.OK()){
17         possibleInputList.add(response.getBody());
18     }else{
19         generateNewWorkflowAfterError();
20         toBeExecutedNode = 0;
21         return;
22     }
23 }
24 toBeExecutedNode++;

```

Figure 12: Microflow enactment algorithm (pseudocode).

Figure 13 shows the interactions when a microflow is enacted. Within a loop, a PUT is used to invoke each virtual microservice for testing purposes, but this would be adjusted for real microservices.

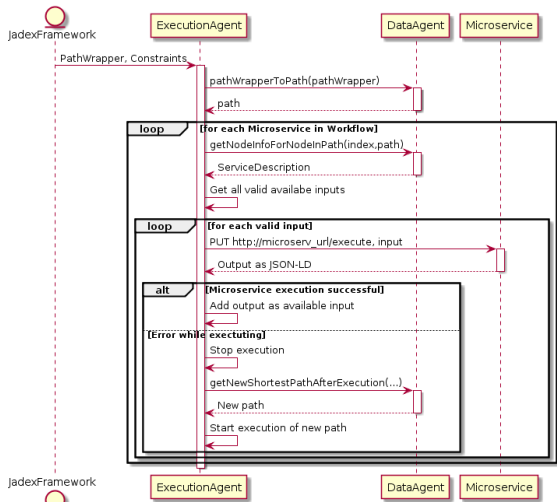


Figure 13: Microflow enactment interactions.

While the service description could be retrieved directly from the microservice, we currently use the internal copy stored during the discovery stage to avoid the additional network and microservice overhead of retrieving this information again. If the description is expected to be highly dynamic, the current description could be retrieved from the microservice during enactment.

5 EVALUATION

To evaluate the solution approach, we investigated if the resource usage of prototype was relatively lightweight, if it could execute the equivalent workflow of a BPMS, and determine if it shows any potential advantage in labor overhead.

The configuration used for the evaluation consisted of a PC with Windows 10 Pro x64, Intel Core i5-4460@3.2 GHz, 8 GB RAM, Java JRE 1.8.0_66-b18. Unless noted, the average of 10 consecutive measurements is given.

5.1 Resource Utilization

To determine the resources utilized by Neo4J, the number of microservices was scaled using 29 different configurations ranging from 100 to 6400 microservices, an extract of which is shown in Table 1. Only one outgoing edge for each microservice was used to keep that variable constant. The drop in RAM usage after 1600 microservices may be a result of garbage collection, and we intend to repeat these measurements with more control over that factor. These measurements show that a very large number of microservices can be supported with relatively little additional RAM or disk impact.

Table 1: Neo4j resource usage.

Number of Microservices	Diskspace (MB)	RAM (MB)
100	0.211	115
400	0.315	120
800	0.716	150
1600	0.741	175
3200	1.28	110
6400	2.39	115

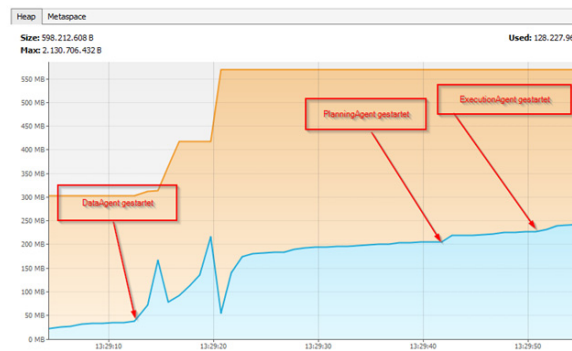


Figure 14: RAM profiling showing agent starting points.

To determine if the Jadex agents have a significant impact on RAM usage, profiling with the VisualVM was performed as shown in Figure 14. The DataAgent was started first (RAM use was in

accordance with Table 1), while the PlanningAgent and ExecutionAgent had no major RAM impacts.

To investigate the performance and scalability of the microflow planning stage, a small program was written that generates z layers of microservices, each layer of which contains m microservices, and each microservice of a layer z has an edge to every microservice of the layer $z+1$.

Neo4j does not explicitly name the algorithm implemented for shortest path, but let us assume it is at least as good as the Dijkstra algorithm, which it also offers and appears to use Fibonacci Heaps (Makrai, 2015), yielding a complexity:

$$O(v \cdot \log v + e) \quad (1)$$

where v are the vertices and e the edges. If n is the number of constraints, then $n+1$ segments have to be computed between the start and end vertex. Let x_i be the possible number of start vertices and y_i the possible number of end vertices for a segment i where $0 \leq i \leq n$, then in every segment there are a maximum of $x_i \cdot y_i$ shortest paths, resulting in:

$$O\left(\sum_{i=0}^n (x_i \cdot y_i) \cdot (v \cdot \log v + e)\right) \quad (2)$$

Thus increasing the number of possible starting or ending nodes has a greater influence on performance.

We performed an experiment comparing the shortest path and Dijkstra algorithm performance, and the shortest path was faster in all cases, so we continued with shortest path.

As expected, when we increased the possible number of starting or ending nodes while keeping the total number of microservices constant, we observed a much larger impact on performance than any increase to the number of segments.

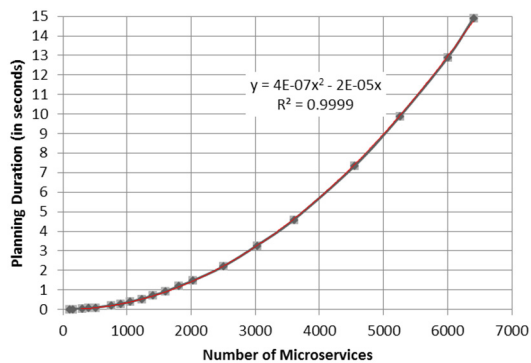


Figure 15: Microflow planning performance of planning duration vs. number of microservices.

Figure 15 shows the planning performance

impact as the total number of microservices increase with a best-fit equation shown. While this may not be ideal, it may suffice for pragmatic usage in non-time critical situations where sufficient CPU resources are available and when the total number of microservices to be considered is limited.

5.2 Microflow Vs. Workflow

To attempt to provide insight into a pragmatic comparison of microflows to standard workflows, a user familiar with our microflow concept and somewhat familiar with BPM, using the microflow described in Section 4.3 and shown in Figure 16 as a basis, and modeled its equivalent as a workflow in AristaFlow BPM Suite (representing a BPMS). The workflow consisted of 12 nodes and 13 edges: Start, Flight Search, Hotel Search, Book Flight, Book Hotel, Booking Error Check (which loops back to start on an error), followed by a conditional Branch to either Pay by CreditCard or Pay by Bank, than a Merge and then an End node.



Figure 16: Microflow shown in Neo4j.

For the microflow, manually preparing the microflow constraints and starting the Jadex and DataAgent involved 4:24 minutes; the automatic planning took 3.9 seconds; and the enactment of the virtual microservices 4.7 seconds. For the equivalent workflow using empty activities that do not actually invoke services, manual creating the process schema took 19:39 minutes while the enactment of the workflow took 8 seconds.

This was not necessarily a "fair" comparison - since a BPMS supports many more capabilities such as correctness checks and the user was not independent. Nevertheless, the point of this exercise was to show that the input constraints needed for microflows (cp. Figure 6) can be more lightweight,

and that the utilization of the dynamic planning capability can reduce the labor overhead of manual planning of workflows in the microservice space, especially if these are expected to vary often. Analogous to the more heavyweight EJB containers vs. more lightweight containers, perhaps the more lightweight form of microflows could be beneficial when the more complete BPMS functionality is not needed.

To validate its exception handling and replanning capability, we manually created situations where certain microservice returned an error, and observed that the agent triggered a replanning consisting of the error segment plus the remaining segments, providing some resilience.

Note that performance was intentionally not optimized in order to provide an indicator of the default viability and investigate how lightweight the approach is. In future work, we plan optimizations.

6 CONCLUSIONS

We described Microflows, an automatic lightweight declarative approach for the workflow-centric orchestration of semantically-annotated microservices using agent-based clients, graph-based methods, and lightweight semantic vocabularies. Microflow principles and its lifecycle were described. Based on a realization, the evaluation showed that the approach is lightweight, while still offering optimization potential. Although its scalability is impeded, depending on the environmental performance constraints and deployment configuration, the automatic planning may be viable for typical workflow scenarios using a limited set of microservices. Further, the evaluation showed that its automated planning offers efficiency benefits vs. manual modelling, and that its enactment performance can be on par with that of commercial BPMS systems.

One advantage we see in the Microflow approach is that the workflow (or plan) is not thoroughly adhoc and dynamic, so that validation and verification checks can be performed before execution and one is assured that the workflow is executable as planned. For instance, if all microservices were there, but a payment service is missing, then a client without this knowledge would work its way through and realize at the very end that it has no way to pay. However, enhanced support for verification and validation of the correctness of the microflow is still needed for users to entrust the automatic planning.

Future work includes integrating advanced verification and validation techniques, optimizing resource usage, integrating semantic support in the discovery service, transactional workflow support, support for gateways, supporting compensation and long-running processes, and enhancing the declarative and semantic support and capabilities.

ACKNOWLEDGEMENTS

The author thanks Florian Sorg for his assistance with the design, implementation, evaluation, and diagrams.

REFERENCES

- Alpers, S., Becker, C., Oberweis, A. and Schuster, T. (2015). Microservice based tool support for business process modelling. In Enterprise Distributed Object Computing Workshop (EDOCW), 2015 IEEE 19th International (pp. 71-78). IEEE.
- Anderson, C., Suarez, I., Xu, Y., & David, K. (2015). An Ontology-Based Reasoning Framework for Context-Aware Applications. In Modeling and Using Context (pp. 471-476). Springer International Publishing.
- Bouguettaya, A., Sheng, Q.Z. and Daniel, F. (2014). Web services foundations. Springer.
- Bratman, M.E., Israel, D.J. and Pollack, M.E. (1988). Plans and resource-bounded practical reasoning. *Computational intelligence*, 4(3), pp.349-355.
- Eureka (2016). Retrieved April 20, 2016 from: <https://github.com/Netflix/eureka/wiki>
- Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures. Doctoral dissertation, University of California, Irvine.
- Florio, L. (2015). Decentralized self-adaptation in large-scale distributed systems. In Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering (pp. 1022-1025). ACM.
- Fowler, M. & Lewis, J. (2014). Microservices a definition of this new architectural term. Retrieved April 15, 2016 from: <http://martinfowler.com/articles/microservices.htm>
- Gartner (2015). Gartner Says Spending on Business Process Management Suites to Reach \$2.7 Billion in 2015 as Organizations Digitalize Processes. Press release. Retrieved April 15, 2016 from: <https://www.gartner.com/newsroom/id/3064717>
- Heitmman, B., Cyganiak, R., Hayes, C. & Decker, S. (2012). An empirically grounded conceptual architecture for applications on the web of data. *Systems, Man, and Cybernetics, Part C: Applications and Reviews*, IEEE Transactions on, 42(1), 51-60.
- IBM (2015). IBM Business Process Manager V8.5.6 documentation. Retrieved May 2, 2016 from:

- http://www.ibm.com/support/knowledgecenter/SSFPJ_S_8.5.6/com.ibm.wbpm.wid.bpel.doc/topics/cprocess_transaction_micro.html
- Karagiannis, G., Jamakovic, A., Edmonds, A., Parada, C., Metsch, T., Pichon, D., ... & Bohnert, T. M. (2014). Mobile cloud networking: Virtualisation of cellular networks. In *Telecommunications (ICT), 2014 21st International Conference on* (pp. 410-415). IEEE.
- Lanthaler, M. (2013). Creating 3rd generation web APIs with hydra. In *Proceedings of the 22nd international conference on World Wide Web companion. International World Wide Web Conferences Steering Committee*, pp. 35-38.
- Lanthaler, M., & Gütl, C. (2012). On using JSON-LD to create evolvable RESTful services. In *Proceedings of the Third International Workshop on RESTful Design* (pp. 25-32). ACM.
- Lanthaler, M. and Gütl, C. (2013). Hydra: A Vocabulary for Hypermedia-Driven Web APIs. In *Proceedings of the 6th Workshop on Linked Data on the Web (LDOW2013) at the 22nd International World Wide Web Conference (WWW2013)*, vol. 996.
- Makrai, G. (2015). Experimenting with Dijkstra's algorithm. Retrieved May 2, 2016 from: <https://gabormakrai.wordpress.com/2015/02/11/experimenting-with-dijkstras-algorithm/>
- Martin, D. et al. (2004). OWL-S: Semantic markup for web services. W3C member submission, 22, pp.2007-04.
- Pokahr, A., Braubach, L., & Lamersdorf, W. (2005). Jadex: A BDI reasoning engine. In *Multi-agent programming* (pp. 149-174). Springer US.
- Rajasekar, A., Wan, M., Moore, R., & Schroeder, W. (2012). Micro-Services: A Service-Oriented Paradigm for. Data Intensive Distributed Computing. In: *Challenges and Solutions for Large-scale Information Management* (pp. 74-93). IGI Global.
- Rao, J. and Su, X. (2004). A survey of automated web service composition methods. In *Semantic Web Services and Web Process Composition* (pp. 43-54). Springer Berlin Heidelberg.
- Sheng, Q. Z. et al. (2014). Web services composition: A decade's overview. *Information Sciences*, 280, 218-238.
- Toffetti, G., Brunner, S., Blöchlinger, M., Dudouet, F., & Edmonds, A. (2015). An architecture for self-managing microservices. In *Proceedings of the 1st International Workshop on Automated Incident Management in Cloud* (pp. 19-24). ACM.
- WFMC (1999). *Workflow Management Coalition: Terminology & Glossary*. WFMC-TC-1011, Issue 3.0.
- Wooldridge, M. (2009). *An introduction to multiagent systems*. John Wiley & Sons.

Scheduling Different Types of Applications in a SaaS Cloud

Georgios L. Stavrinides and Helen D. Karatza

Department of Informatics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece
{gstavrin, karatza}@csd.auth.gr

Keywords: Scheduling, Complex Workload, SaaS Cloud, Performance, Modeling, Simulation.

Abstract: As Software as a Service (SaaS) cloud computing gains momentum, the efficient scheduling of different types of applications in such platforms is of great importance, in order to achieve good performance. In SaaS clouds the workload is usually complex and comprises applications with various degrees of parallelism and priority. Therefore, one of the major challenges is to cope with the case where high-priority real-time single-task applications arrive and have to interrupt other non-real-time parallel applications in order to meet their deadlines. In this case, it is required to effectively deal with the real-time applications, at the smallest resulting degradation of parallel performance. In this paper, we investigate by simulation the performance of strategies for the scheduling of complex workloads in a SaaS cloud. The examined workload consists of non-real-time applications featuring fine-grained parallelism (gangs) and periodic high-priority soft real-time single-task applications that can tolerate deadline misses by bounded amounts. We examine the impact of gang service time variability on the performance of the scheduling algorithms, by considering service demands that follow a hyper-exponential distribution. The simulation results reveal that the relative performance of the employed scheduling strategies depends on the type of the workload.

1 INTRODUCTION

Software as a Service (SaaS) cloud computing has become prevalent in recent years, replacing the traditional software delivery model, which required the installation of the software on the client's servers. According to this emerging software distribution approach, which may be considered as the evolution of the Service-Oriented Architecture (SOA) model, the software is hosted by the vendor and made available to the end-users over the Internet, as a cloud service (Cusumano, 2010) and (Beloglazov et al., 2012).

Some of the main benefits of the SaaS cloud computing model are:

- The software is always up to date.
- The software can be accessed via various devices and from different locations.
- It provides overall easier administration and maintenance, as the SaaS clients do not have to acquire, maintain and monitor expensive hardware and software infrastructure.

The clients use the software services, without any control on the host environment, either on a pay-as-you-go basis or a subscription based pricing model (Bittencourt et al., 2012) and (Dillon et al., 2010).

1.1 Motivation

SaaS cloud computing usually relies on a *multi-tenant* model, where applications of different users run on the same virtual machines (VMs). Due to the tremendous increase of users and the number of different applications sharing the underlying virtualized resources, the performance of SaaS clouds has become a crucial area of research. Therefore, the efficient scheduling of different types of applications in such platforms is of great importance, in order to effectively utilize the underlying multi-tenant infrastructure and achieve good performance, while maintaining a certain level of *Quality of Service (QoS)* (Hofer and Karagiannis, 2011), (Rimal et al., 2009) and (Stavrinides and Karatza, 2015).

In SaaS clouds the workload is usually complex and comprises applications with various degrees of parallelism and priority. Consequently, one of the major challenges is to cope with the case where high-priority real-time single-task applications arrive and have to interrupt other non-real-time parallel applications in order to meet their deadlines. In this case, it is required to effectively deal with the real-time applications, at the smallest resulting degradation of parallel performance. Moreover, part of the workload may consist of fine-grained parallel

applications that present high variability in their service times. In this case, the scheduling algorithm should cope with bursts of very small service times and a few but very large ones, compared to the average service time of applications.

1.2 Contribution

In this paper, we focus on the performance of strategies for the scheduling of complex workloads in a SaaS cloud with multi-tenant VMs. The examined workload consists of non-real-time applications featuring fine-grained parallelism and periodic high-priority soft real-time single-task applications that can tolerate deadline misses by bounded amounts.

We examine the impact of service time variability of parallel applications on the performance of the scheduling algorithms, by considering service demands that follow a hyper-exponential distribution. To our knowledge, scheduling complex workloads, including fine-grained parallel applications and periodic soft real-time single-task jobs, in such a framework, does not appear elsewhere in the research literature.

1.3 Background and Related Work

One type of workload submitted to SaaS clouds is *bag-of-tasks (BoT)* applications. Each BoT is a collection of independent tasks that do not communicate with each other and can run in any order (Karatzas, 2004), (Kim et al., 2007), (Moschakis and Karatzas, 2015), (Papazachos and Karatzas, 2015) and (Terzopoulos and Karatzas, 2016). Another type of workload is applications that consist of interdependent tasks, which often have precedence constraints among them and thus form a *directed acyclic graph (DAG)* (Stavrinides and Karatzas, 2010), (Stavrinides and Karatzas, 2011), (Stavrinides and Karatzas, 2012) and (Stavrinides and Karatzas, 2014).

However, most often the workload in SaaS clouds consists of fine-grained parallel applications. They consist of frequently communicating tasks, which are scheduled to run simultaneously on different VMs, as *gangs*. Gang scheduling is an efficient resource management technique in the case where parallel applications feature fine-grained parallelism.

Since with gang scheduling a gang's task can start execution only if all of the other sibling tasks can also start processing, some VMs may remain idle even when there are tasks waiting in their queues. This problem becomes more complex when

the workload also includes real-time applications which have higher priority than gangs.

Gang scheduling algorithms have been proposed and studied by many authors, each differing in the way resources are shared among the jobs (Stavrinides and Karatzas, 2008), (Stavrinides and Karatzas, 2009), (Streit, 2005) and (Zhang et al., 2003).

In (Karatzas, 2006), the performance of two well-known gang scheduling methods is studied, the *Adapted-First-Come-First-Served (AFCFS)* and the *Largest-Job-First-Served (LJFS)*. It has been shown that in many cases LJFS performs better than AFCFS. However, in this case no real-time jobs are considered and the overall performance is expressed by the average response time of gangs.

In this paper, we study gang scheduling in a queueing network model of a SaaS cloud. The performance of two policies that are variations of AFCFS and LJFS, is studied under various workloads, which include gangs, as well as periodic real-time single-task jobs that have higher priority than gangs. The periodic single-task jobs are considered as soft-real time applications, in the sense that they can tolerate deadline misses by bounded amounts.

Related research includes (Karatzas, 2007) and (Karatzas, 2008). These papers examine critical sporadic jobs that need to start execution upon arrival and therefore they interrupt gangs. In this paper, we examine periodic soft real-time jobs that can tolerate some delay called *slack time*. This happens in the case when gangs in service require only a small amount of time to finish their execution that is less than or equal to the slack time.

Scheduling workloads consisting of soft-real time jobs and gangs in distributed systems has also been studied in (Karatzas, 2014). However, the workload considered in this case does not include parallel applications with highly variable service demands.

Tardiness bounds for sporadic real-time task systems have been studied in (Devi and Anderson, 2006) and (Leontyev and Anderson, 2010). However, these papers do not consider gangs in their workloads.

In this paper, we examine the impact of gang service time variability on the performance of the scheduling algorithms, by considering service demands that follow a hyper-exponential distribution. A high variability in task service demand implies that there is a proportionately large number of service demands that are very small compared to the mean service time and a comparatively small number of service demands that

are very large.

When a gang with a long service demand starts execution, it occupies its assigned VMs for a long time interval, and depending on the scheduling policy that is employed, it may introduce inordinate queueing delays for other tasks waiting for service.

The performance evaluation of complex distributed systems such as clouds is often possible only by simulation rather than by analytical techniques, due to the complexity of the systems. Simulation can provide important insights into the efficiency and tradeoffs of scheduling in such environments. Therefore, due to the complexity of the system under study, we use discrete event simulation to evaluate the performance of the scheduling algorithms.

The remainder of this paper is organized as follows: Section 2 introduces the model and the methodology employed, describing the system and workload models. Sections 3 and 4 describe the routing and gang scheduling policies respectively, whereas Section 5 presents the performance metrics. The model implementation and its input parameters, as well as the simulation results, are presented and analysed in Section 6. Finally, Section 7 summarizes the paper and provides future research directions.

2 MODEL AND METHODOLOGY

2.1 System and Workload Models

This paper uses a simulation model to address performance issues. The target SaaS cloud is considered to consist of a set V of p virtual machines $V = \{VM_1, VM_2, \dots, VM_p\}$, that are fully connected by a virtual network (Figure 1).

It is assumed that the communication between the virtual machines is contention-free. Each virtual machine VM_i serves its own queue of tasks and has mean execution rate μ . The VMs in the cloud are multi-tenant, that is each VM processes tasks of applications submitted by different users.

Each parallel job is submitted for execution in the cloud by a user. Users share the virtual machines to run concurrently their applications. Since the parallel applications are submitted dynamically by multiple users, it is assumed that they arrive in the cloud in a Poisson stream with mean arrival rate λ .

The number of tasks in a parallel job x is the job's *degree of parallelism* and it is represented as $t(x)$. In Figure 1 virtual machines are allocated to a parallel job x , which has j parallel tasks. VM_p is assigned to a periodic single-task job.

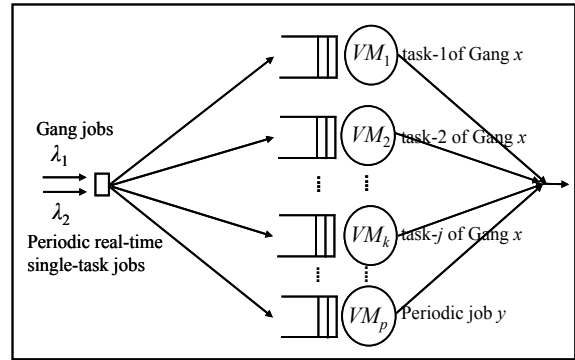


Figure 1: The queueing network model.

If $vm(x)$ represents the number of VMs required by job x , then the following relationship holds: $1 \leq t(x) = vm(x) \leq p$. We call *size* of job x the number of its parallel tasks. A job is *small* (*large*) if it consists of a small (*large*) number of tasks. Soft real-time jobs are periodic single-task jobs.

Each task of a gang x is routed to a different VM. Gang x starts to execute only if all of the $vm(x)$ VMs assigned to its tasks are available. Otherwise, all of the x gang tasks wait in their assigned queues. When a gang terminates execution, all VMs assigned to it are released.

An important issue that arises is the need to serve a real-time single-task job on a virtual machine executing a gang task. Therefore, a mechanism is needed to deal with the fact that resources allocated to a gang are changing. The scheduler must give priority to real-time applications, but at the same time provide good performance for parallel applications that compete for resources.

For each soft real-time application there is a slack time during which it can wait for a gang that needs a small amount of time to finish execution. The slack time is a constant tardiness that is independent of elapsed time. If the time until the completion of a gang is larger than the slack time, then the job scheduler interrupts the gang upon the arrival of a soft real-time application, so that it can occupy a VM. The remaining virtual machines that are assigned to the interrupted gang can serve tasks of other gangs that are waiting at their queues. All of the work that was performed on all tasks associated with the interrupted gang must be redone. The tasks of an interrupted parallel application are rescheduled for execution at the head of their assigned queues.

A technique used to evaluate the performance of the scheduling disciplines, is experimentation using a synthetic workload. The workload considered here is characterized by the following parameters:

- The distribution of the number of tasks of gangs.

- The distribution of gang tasks service demand.
- The distribution of soft real-time jobs service demand.
- The mean inter-arrival time of gangs.
- The period of soft real-time jobs.

2.2 The Distribution of Gang Sizes

We assume that the number of gang tasks is uniformly distributed in the range $[1, p]$. Therefore, the mean number of tasks per parallel application is equal to $m = (1+p)/2$.

2.3 Service Time Distribution

For the service demands of parallel applications and real-time jobs we assume that:

- Service demands of gang tasks follow a hyper-exponential distribution with a coefficient of variation CV and a mean $1/\mu$. CV determines the degree of variability. This is the ratio of the standard deviation of task execution time to its mean. $CV > 1$ implies a large variability, compared to the variability of the exponential distribution.
- Real-time jobs service demands are exponentially distributed with a mean of $1/\mu$.

2.4 Distribution of Job Inter-arrival Times

We consider two arrival streams, one for gangs and one for real-time jobs:

- The inter-arrival times of gangs are exponential random variables with a mean of $1/\lambda_1$.
- The inter-arrival times of soft real-time jobs are constant time intervals equal to $1/\lambda_2$ (i.e. the real-time jobs are periodic).

3 ROUTING POLICIES

Job routing in SaaS clouds is an important aspect for good performance. In this paper, our workload model consists of two different types of jobs. Therefore, we examine two different routing policies, one for each job type.

The first job type concerns the parallel applications, which are gangs. Hence, in this case routing occurs at the task level. The second job type concerns the single task real-time applications. Consequently, in this case routing occurs at the job level.

3.1 Parallel Applications Routing

A variation of the *join the shortest queue* policy is used. That is, the $t(x)$ tasks of a gang x are assigned to the shortest $t(x)$ of the p queues, every task to a different VM queue.

3.2 Real-time Applications Routing

The routing policy of real-time jobs is *join the shortest queue*. In the experiments that we have conducted for this research, appropriate values of the period have been chosen, so that there are no further real-time job arrivals in the case where all of the VMs serve real-time jobs.

4 PARALLEL APPLICATIONS SCHEDULING POLICIES

It is assumed that the scheduler knows the exact number of VMs required by each gang. The following two scheduling strategies are employed in our simulations.

4.1 Adapted-First-Come-First-Served with Execution Interruption based on Slack Time (AFCFS-ST)

This strategy schedules a gang job whenever VMs assigned to its tasks are available. When there are not enough VMs available for a large job whose tasks are waiting in the front of the queues, then the AFCFS-ST policy schedules tasks of smaller jobs in the queues. A problem with this scheduling policy is that it tends to favor small gangs at the expense of larger gangs and thus it may increase system fragmentation. A gang job may be interrupted upon arrival of a real-time single-task job.

4.2 Largest-Job-First-Served with Execution Interruption based on Slack Time (LJFS-ST)

With this strategy tasks that belong to larger gangs are placed at the head of queues. All gang tasks in queues are searched in order, and the first jobs whose assigned VMs are available begin execution. This scheduling method favors large, highly parallel gangs at the expense of smaller gangs, but in many cases this treatment of large gangs is acceptable. For example, supercomputers are often used to particularly run highly parallel jobs for fast

execution. However, LJFS-ST involves an overhead because the VM queues need to be re-arranged each time a new gang is added. Similarly to the AFCFS-ST case, a gang job may be interrupted upon arrival of a real-time job.

In both cases of AFCFS-ST and LJFS-ST, when a real-time job terminates, the interrupted gang may not resume execution immediately, as some of the VMs assigned to its tasks may be running other jobs. Those jobs will not terminate at the same time so the interrupted gang will not use the assigned VMs efficiently. It is worth noting that when a real-time job is assigned to a virtual machine, it is not only the gang tasks of the particular VM queue that are delayed, but also gang tasks in other queues that have a sibling task waiting in the particular queue.

5 PERFORMANCE METRICS

The response time rt_i of a gang i is the time interval from the dispatching of its tasks to different VM queues, to the service completion of the gang – this is the time spent in VM queues plus the time spent in service.

The average response time RT of n gangs is defined as:

$$RT = \frac{\sum_{i=1}^n rt_i}{n} \quad (1)$$

Additionally, we weight each gang's response time with its size (Streit, 2005). Consequently, it is avoided that gangs with the same execution time, but with different number of tasks, have the same impact on performance. The average weighted response time WRT of n gangs is defined as:

$$WRT = \frac{\sum_{i=1}^n vm(x_i) \times rt_i}{\sum_{i=1}^n vm(x_i)} \quad (2)$$

The parameters used in the simulation experiments are shown in Table 1.

Table 1: System parameters and performance metrics.

ST	Slack time of real-time jobs
p	Number of VMs
m	Average number of tasks per parallel job
μ	Mean execution rate of a VM
$1/\mu$	Mean execution time of a VM

CV	Coefficient of variation of service demands
λ_1	Mean arrival rate of gangs
$1/\lambda_2$	Period of real-time jobs
U	Average VM utilization
RT	Average response time of gangs
D_{RT}	Relative (%) decrease in RT when LJFS-ST method is employed instead of the AFCFS-ST policy
WRT	Average weighted response time of gangs
D_{WRT}	Relative (%) decrease in WRT when LJFS-ST method is employed instead of the AFCFS-ST policy

6 SIMULATION RESULTS AND DISCUSSION

6.1 Model Implementation and Input Parameters

The queueing network model described in Section 2 is implemented via discrete event simulation. Due to the complexity of the system and the workload model under study, we implemented our own discrete event simulation program in C, tailored to the requirements of the specific case study.

In order to derive the mean values of the performance parameters we used the independent replications method. For each set of workload parameters we run 30 replications of the simulation with different seeds of random numbers and for 32,000 served jobs in each replication. We considered this simulation length long enough to derive results, as we found by experimentation that longer runs did not affect simulation output significantly. The use of sufficiently long simulation runs, is one of the ways to reduce the effect of initial bias on simulation results.

For every mean value of the performance parameters, a 95% confidence interval was evaluated. The half-widths of all confidence intervals are less than 5% of their respective mean values evaluated. For our experiments, we used the simulation input parameters shown in Table 2.

In our workload model there are on average $m = (p+1)/2 = 8.5$ tasks per parallel job. Therefore, if we do not consider any real-time jobs and all VMs are busy serving gangs, then an average of $p / m = 1.88235$ parallel jobs can be served per each unit of time. This implies that we should choose a λ_1 such that the condition $\lambda_1 < 1.88235$ holds, so that the VM queues will not be saturated. However, due to the real-time job arrivals, the number of VMs that are

available for gang service is $[p - (\lambda_2/\mu)]$. Therefore, we have to choose a value of λ_1 for which the following relationship holds:

$$\lambda_1 < [p - (\lambda_2/\mu)] / m \quad (3)$$

However, due to gang scheduling, there are often idle VMs, despite the fact that there are gang tasks waiting in the respective queues. Therefore, the VM queues get very easily saturated when the mean inter-arrival time of gangs is close to $[p - (\lambda_2/\mu)] / m$. After experimental runs with various values of $1/\lambda_1$, we chose 0.78 as the smallest mean inter-arrival time of gangs for the experiments, for all cases of $1/\lambda_2$. The input parameter values which were used for the experiments are presented in Table 2.

Table 2: Input parameters.

ST	0.1
$1/\mu$	1
$1/\lambda_1$	0.84, 0.78
$1/\lambda_2$	20, 30, 40
CV	2

6.2 Performance Analysis

The simulation results that are presented next describe the relative performance of the two gang scheduling policies.

6.2.1 Performance with Regard to Parallel Applications Service

Figure 2 presents D_{RT} versus $1/\lambda_1$ in the cases of period $1/\lambda_2 = 20, 30$ and 40 . Figure 3 presents D_{WRT} versus $1/\lambda_1$ in the cases of period $1/\lambda_2 = 20, 30$ and 40 .

Figures 2 and 3 show that for both arrival rates of gangs and for all cases of real-time jobs period, the LJFS-ST method yields lower mean response time than AFCFS-ST.

In each real-time job period case, the superiority of LJFS-ST over AFCFS-ST is more significant in the case of high load ($1/\lambda_1 = 0.78$). This is due to the fact that the advantages of the LJFS-ST case are better exploited when there is a sufficient number of gangs in the queues, so that they can be selected according to the LJFS-ST criteria.

The simulation results also reveal that the D_{RT} increase with increasing load is more significant in the larger periods. This may be explained by the fact that the smaller the period of real-time jobs, the larger is the possibility that fewer virtual machines are available for gang service. Therefore, the potential of the LJFS-ST policy is not completely

exploited, as large applications cannot find enough VMs available to serve them.

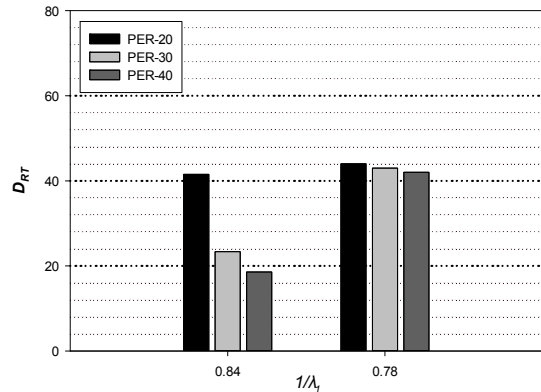


Figure 2: D_{RT} ratio versus $1/\lambda_1$, soft real-time jobs with period 20, 30 and 40.

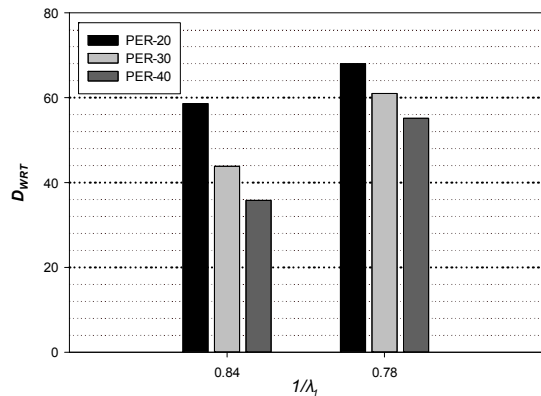


Figure 3: D_{WRT} ratio versus $1/\lambda_1$, soft real-time jobs with period 20, 30 and 40.

In each case of the arrival rate of gangs, D_{RT} decreases with increasing period size. This is because when real-time jobs arrive less frequently, then a smaller number of gangs are interrupted. The D_{RT} decrease with increasing period size is smaller in the case of heavier load ($1/\lambda_1 = 0.78$). This is due to the fact that when a real-time job blocks a gang, then it does not only affect the performance of the blocked gang, but it also affects the performance of a number of subsequent gangs.

Figure 3 shows that the observations that hold for the relative performance of the scheduling policies in terms of the average response time of parallel applications, also generally hold in terms of the average weighted response time. In all cases D_{WRT} is larger than D_{RT} . Therefore, the superiority of LJFS-ST over AFCFS-ST appears more significant when the job response time is weighted by the degree of parallelism of the jobs.

6.2.2 Performance with Regard to VM Utilization

Tables 3 and 4 depict the mean virtual machine utilization U versus $1/\lambda_1$, for period size $1/\lambda_2 = 20, 30$ and 40 respectively. In most cases, the mean VM utilization is either equal for both of the scheduling policies or it is very slightly larger in the LJFS-ST case. This is because the LJFS-ST policy schedules large gangs first and consequently, it schedules jobs on the available VMs more efficiently than the AFCFS-ST policy.

Table 3: U for AFCFS-ST.

$1/\lambda_1$	0.84	0.78
PER-20	0.711	0.740
PER-30	0.672	0.716
PER-40	0.636	0.690

Table 4: U for LJFS-ST.

$1/\lambda_1$	0.84	0.78
PER-20	0.712	0.768
PER-30	0.672	0.718
PER-40	0.637	0.695

However, with both scheduling strategies, part of the virtual machines utilization is comprised of repeated gang work due to the real-time job service. The amount of the repeated work depends on the number of gang tasks, the service demand of each task and the work that has already be done by the gang at the moment of interruption. It is possible for a gang to be interrupted several times during its execution. This would be caused by multiple real-time job arrivals at different virtual machines serving tasks of the same gang.

For each λ_1 , the utilization slightly decreases with increasing period size. This is due to the fact that it is more possible to have to restart the parallel applications execution when the period of real-time applications is small, than when it is large. However, due to the variability in gang service demand in our workload model, only the gangs that have very large service demands may experience multiple service interruptions.

7 CONCLUSIONS AND FURTHER RESEARCH

This paper examines the performance of two gang scheduling policies, the LJFS-ST and the AFCFS-ST, in a SaaS cloud in the presence of periodic soft

real-time single-task jobs. The objective is to enhance the performance of parallel applications, assuming that soft real-time jobs can tolerate a delay.

Simulation results show that the relative performance of the two scheduling methods depends on the workload. With regard to the performance metrics considered in this paper, the LJFS-ST method performs better than the AFCFS-ST, in all cases of workloads, providing promising results.

Furthermore, the superiority of LJFS-ST over AFCFS-ST appears more significant when the response time of parallel applications is weighted by their degree of parallelism.

Our future research plans include the examination of cases where along with gangs there are also workflow applications. Moreover, we plan to consider additional distributions for service demands and investigate their impact on the monetary cost charged to the end-users of the SaaS cloud.

REFERENCES

- Cusumano, M., 2010. Cloud computing and SaaS as new computing platforms. *Communications of the ACM*. ACM, 53(4), 27-29.
- Beloglazov, A., Abawajy, J. and Buyya, R., 2012. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Generation Computer Systems*. Elsevier, 28(5), 755-768.
- Bittencourt, L. F., Madeira, E. R. M. and Da Fonseca, N. L. S., 2012. Scheduling in hybrid clouds. *Communications Magazine*. IEEE, 50(9), 42-47.
- Devi, U. C. and Anderson, J. H., 2006. Flexible tardiness bounds for sporadic real-time task systems on multiprocessors. In *IPDPS'06, 20th IEEE International Parallel and Distributed Processing Symposium*. IEEE, Rhodes Island, Greece.
- Dillon, T., Wu, C. and Chang, E., 2010. Cloud computing: issues and challenges. In *AINA'10, 24th IEEE International Conference on Advanced Information Networking and Applications*. IEEE, Perth, Australia, pp. 27-33.
- Hofer, C. N. and Karagiannis, G., 2011. Cloud computing services: taxonomy and comparison. *Journal of Internet Services and Applications*. Springer, 2(2), 81-94.
- Karatza, H. D., 2004. Simulation study of multitasking in distributed server systems with variable workload. *Simulation Modelling Practice and Theory*. Elsevier, 12(7), 591-608.
- Karatza, H. D., 2006. Scheduling gangs in a distributed system. *International Journal of Simulation: Systems, Science Technology*. UK Simulation Society, 7(1), 15-

- 22.
- Karatzas, H. D., 2007. Performance of gang scheduling policies in the presence of critical sporadic jobs in distributed systems. In *SPECTS'07, 2007 International Symposium on Performance Evaluation of Computer and Telecommunication Systems*. SCS, San Diego, CA, pp. 547-554.
- Karatzas, H. D., 2008. The impact of critical sporadic jobs on gang scheduling performance in distributed systems. *Simulation: Transactions of the Society for Modeling and Simulation International*. Sage Publications, 84(2-3), 89-102.
- Karatzas, H. D., 2014. Scheduling Jobs with different characteristics in distributed systems. In *CITS'14, 2014 International Conference on Computer, Information and Telecommunication Systems*. IEEE, Jeju Island, South Korea, pp. 1-5.
- Kim, K. H., Buyya, R. and Kim, J., 2007. Power aware scheduling of bag-of-tasks applications with deadline constraints on DVS-enabled clusters. In *CCGRID'07, 7th IEEE International Symposium on Cluster Computing and the Grid*. IEEE, Rio de Janeiro, Brazil, pp. 541-548.
- Leontyev, H. and Anderson, J. H., 2010. Generalized tardiness bounds for global multiprocessor scheduling. *Real-Time Systems*. Springer, 44(1-3), 26-71.
- Moschakis, I. A. and Karatzas, H. D., 2015. A meta-heuristic optimization approach to the scheduling of Bag-of-Tasks applications on heterogeneous Clouds with multi-level arrivals and critical jobs. *Simulation Modelling Practice and Theory*. Elsevier, 57, 1-25.
- Papazachos, Z. C. and Karatzas, H. D., 2015. Scheduling bags of tasks and gangs in a distributed system. In *CITS'15, 2015 International Conference on Computer, Information and Telecommunication Systems*. IEEE, Gijón, Spain, pp. 1-5.
- Rimal, B. P., Choi, E. and Lumb, I., 2009. A taxonomy and survey of cloud computing systems. In *NCM'09, 5th International Joint Conference on INC, IMS and IDC*. IEEE, Seoul, Korea, pp. 44-51.
- Stavriniades, G. L. and Karatzas, H. D., 2008. In *SPECTS'08, 2008 International Symposium on Performance Evaluation of Computer and Telecommunication Systems*. IEEE, Edinburgh, UK, pp. 1-7.
- Stavriniades, G. L. and Karatzas, H. D., 2009. Fault-tolerant gang scheduling in distributed real-time systems utilizing imprecise computations. *Simulation: Transactions of the Society for Modeling and Simulation International*. Sage Publications, 85(8), 525-536.
- Stavriniades, G. L. and Karatzas, H. D., 2010. Scheduling multiple task graphs with end-to-end deadlines in distributed real-time systems utilizing imprecise computations. *Journal of Systems and Software*. Elsevier, 83(6), 1004-1014.
- Stavriniades, G. L. and Karatzas, H. D., 2011. Scheduling multiple task graphs in heterogeneous distributed real-time systems by exploiting schedule holes with bin packing techniques. *Simulation Modelling Practice and Theory*. Elsevier, 19(1), 540-552.
- Stavriniades, G. L. and Karatzas, H. D., 2012. Scheduling real-time DAGs in heterogeneous clusters by combining imprecise computations and bin packing techniques for the exploitation of schedule holes. *Future Generation Computer Systems*. Elsevier, 28(7), 977-988.
- Stavriniades, G. L. and Karatzas, H. D., 2014. The impact of resource heterogeneity on the timeliness of hard real-time complex jobs. In *PETRA'14, 7th International Conference on Pervasive Technologies Related to Assistive Environments*. ACM, Rhodes Island, Greece, pp. 65:1-65:8.
- Stavriniades, G. L. and Karatzas, H. D., 2015. A cost-effective and QoS-aware approach to scheduling real-time workflow applications in PaaS and SaaS clouds. In *FiCloud'15, 3rd International Conference on Future Internet of Things and Cloud*. IEEE, Rome, Italy, pp. 231-239.
- Streit, A., 2005. Enhancements to the decision process of the self-tuning dynP scheduler. In *JSSPP'05, 11th Workshop on Job Scheduling Strategies for Parallel Processing*. Springer, Cambridge, MA, pp. 63-80.
- Terzopoulos, G. and Karatzas, H. D., 2016. Bag-of-Tasks load balancing on power-aware clusters. In *PDP'16, 24th Euromicro International Conference on Parallel, Distributed and Network-Based Processing*. IEEE, Heraklion, Crete.
- Zhang, Y., Franke, H., Moreira, J. and Sivasubramaniam, A., 2003. An integrated approach to parallel scheduling using gang-scheduling, backfilling and migration. *IEEE Transactions on Parallel and Distributed Systems*. IEEE, 14(3), 236-247.

On Service e-Marketplaces for Independent and Assisted Living

Leszek A. Maciaszek

Institute of Business Informatics, Wroclaw University of Economics, Poland

leszek.maciaszek@ue.wroc.pl

Keywords: e-Marketplace, e-Service, Independent Living, Ambient Assisted Living, Business Models, Service-Oriented Software Engineering, Design Science.

Abstract: e-Marketplaces for services are governed by similar, but not the same business and technology principles as e-Marketplaces for products. Additionally, in the particular case of e-Marketplaces for senior citizens, service innovation must be requestor-focused. Immobile (residential, fixed) and mobile ambient (sensors, actuators) intelligence devices cover the spectrum of requestor (consumer, patient) monitoring (daily living, health status, intensive care). On the society side, the aim is to enable older persons, nearing or in retirements, healthy or unhealthy, to live socially-included and economically-active lives. On the IT side, the aim is to provide innovative systems and solutions embracing contemporary shifts of computing paradigm to service science, cloud computing, mobile connectivity, business processes, and societal participation. In this paper we report on selected business and technological implications of embarking on the development of a service e-Marketplace platform for independent and assisted living. We discuss plausible business models, identify major success factors, and present software engineering challenges underpinning development of such platforms.

1 INTRODUCTION

Physical and psychological health of older persons brings concrete social and economic benefits and it is an important enabling factor for the sustained economic growth in the world of otherwise unsustainable-economically demographic change. The increased participation of older persons, while demanded by the demographic change, is very much in line with the observed transition to a decentralized and skill-based service economy. The experience, knowledge and wisdom of senior citizens are a huge asset and underutilized enabler of Internet-facilitated service economy of today. The understanding and explanation of these complex relationships constitutes the social and economic background for presented research.

Globally, life expectancy for people born between 2010 and 2015 is 70 years (77 in Europe) and it is expected to keep rising. Moreover, population aged 60 or over accounts in 2015 for 12% of the global population (24% in Europe) and this age group is the fastest growing (3.26% per year) (UN, 2015).

The demographic changes have many economic and social implications. On one hand, senior citizens are healthier and more active than past generations.

This impacts on the lifestyle needs of senior citizens. On the other hand, longevity is associated with long-lasting medical conditions and disabilities. This impacts on the healthcare needs of seniors. As reported in Pritchard and Potter (2011) and elsewhere, the increasing life expectancies and associated healthcare expenditures are becoming unsustainable for world economies. To sustain the pension and healthcare growing costs, there is a need to activate older persons by providing opportunities for greater economic and social engagement and to extend care from hospitals, nursing accommodations, etc. to the home surroundings of consumers/patients. With the assistance of tele-monitoring and tele-treatment, health management can contribute to the benefits and become a part of life engineering (Oesterle, 2014; Hess et al., 2014).

From the technology perspective, the first discourse is termed Independent Living (IL), the second – Ambient Assisted Living (AAL). In both cases, the battle ground is to deliver appropriate services to older persons. Information technology plays here an ever increasing role, including through provision of e-Marketplaces for products and services of interest to senior citizens.

The paper has the following structure. The next Section discusses the very concept of service e-Marketplace. In Section 3, we describe business models for service e-Marketplaces and we list the factors for making the platform a success. In Section 4, we address selected software engineering challenges faced by the developers of a service e-Marketplace. In the Conclusions Section, we place our research in the wider context of the innovation chain in service e-Marketplace R&D.

2 SERVICE e-MARKETPLACES

e-Marketplaces for services (such as Airbnb, OpenTable, or BlablaCar) are governed by different value propositions, business models and technology principles than e-Marketplaces for products (such as Alibaba, eBay, or MercadoLivre). The differences stem from differences between products and services. A requestor of a product request a tangible thing and takes the ownership of that thing when provided by the supplier. A requestor of a service requests an action without taking the ownership of the service.

This said, we must recognize that most services have an element of products in them and vice versa. The dichotomy between these two concepts has been replaced by a service-product continuum (Targowski, 2009). On one hand, products are servitized; on the other hand, services are productized (Cusumano, 2008).

In the context of R&D reported in this paper, we must distinguish between a service and an IT service. We follow here the definitions of ITIL (IT Infrastructure Library) (Aguter, 2013, p.17):

- "A service is a means of delivering value to customers by facilitating outcomes customers want to achieve without the ownership of specific costs and risks."
- "A service provided by an IT service provider. An IT service is made up of a combination of information technology, people and processes. A customer facing IT service directly supports the business processes of one or more customers, and its service level targets should be defined in a Service Level Agreement. Other IT services, called supporting services, are not directly used by the business, but are required by the service provider to deliver customer-facing services."

From the IT perspective, vendors of traditional "boxed" software products use the cloud as a means of servitizing the product (and offering it without changing the ownership); on the other hand,

productized services (i.e. automation of services, such as movies over Internet) enable "people to participate in a growing number of service-related activities without having to be physically present" (Targowski, 2009, p.57).

e-Marketplaces differ significantly from traditional e-Commerce systems. Firstly, they reinstate the function of intermediaries between the requestor/buyer and supplier/seller (i.e. the function that traditional e-Commerce systems can do without). Secondly, and more importantly, e-Marketplaces connect actors in many-to-many collaborative networks, in which diverse offerings of suppliers are facilitated by diverse intermediaries and meet with diverse realizations of requestors. Such systems emphasize the appointment coordination process between the three parties. We define the appointment coordination as "a process, which leads to establishing time, locations, participants and goal of the appointment" (Biziel et al., 2016).

3 BUSINESS MODELS AND SUCCESS FACTORS

Service innovation is requestor-focused. Requestor market, as the primary driver of service e-Marketplace innovation, challenges the way companies innovate and evolve with IT. The innovation ideas need to tap into the phenomenon of consumerization and personalization as the tendency for new IT solutions to emphasize requestor-focused service provision and to emerge first in the personal requestor market and then spread into business and government organizations. Consumerization opens up an opportunity for new business models and ways of value creation.

The shifts to consumerization and personalization underlies next generation service e-Marketplace solutions, in which service requestor is placed in the centre of attention, while the service supplier is given enough incentive to use the e-Marketplace (and, in most business models, pay for the use of it). The formula of consumer being a core entity in lifestyle engineering domain matches well the traditional marketing principle that "the customer is always right".

A notable exception has traditionally been the public sector, in particular healthcare services. In the health management domain, the service supplier (physician) has supreme power over the service requestor (patient), who is seen as a grateful, passive entity. Next generation service e-marketplaces bring

about a wave of change. The road is bumpy, though, for reasons of (1) professional autonomy of health service providers and (2) perception that resource utilization in publicly-funded health services constitutes public rather than private (patient) good (Laing and Hogg, 2002).

The business models need to consider three dimensions: economic, operational and strategic (Osl et al., 2008). Issues to be considered involve key partners, suppliers, resources and activities in the e-Marketplace venture. Value propositions, cost structure, revenue streams need to be discussed and defined. The establishment of customer relationships and communication channels to various customer segments are fundamental to the business success.

The models for the use of a service e-Marketplace platform involve many actors. A vendor of a cloud Software-as-a-Service (SaaS) solution offers tenant instances to suppliers of concrete e-Marketplace services. Suppliers/intermediaries act as context integrators for interested service providers (e.g. a hospital). A provider enlists service performers (e.g. a nurse) as subscribers to a cloud-based service e-Marketplace platform. A performer delivers the

service to a requestor (e.g. a patient).

Figure 1 presents actors and revenue streams for the service e-Marketplace business case. Requestors (consumers, patients) solicit services of Suppliers by using service e-Marketplace system. Services can be obtained via Intermediaries, who have access to service context and information about requestors' standing. They can act as concierge or call centres and can make appointments for services on behalf of Providers. Alternatively, Requestors can make direct appointments with Providers via the service e-Marketplace platform. In a typical situation, Provider employs Performers, who actually execute a service.

The Intermediary aspect of e-Marketplace platform enables human-driven concierge-like appointment coordination made possible by all communication means (Internet, telephone, email, etc.). However, the principal modus-operandi of the platform is to encourage Requestors and Suppliers to manage appointment and service provisioning with web-based applications using context-aware and purpose-designed mobile and non-mobile devices. In some cases, in particular in e-health domain, delivery of services can be auto-activated by

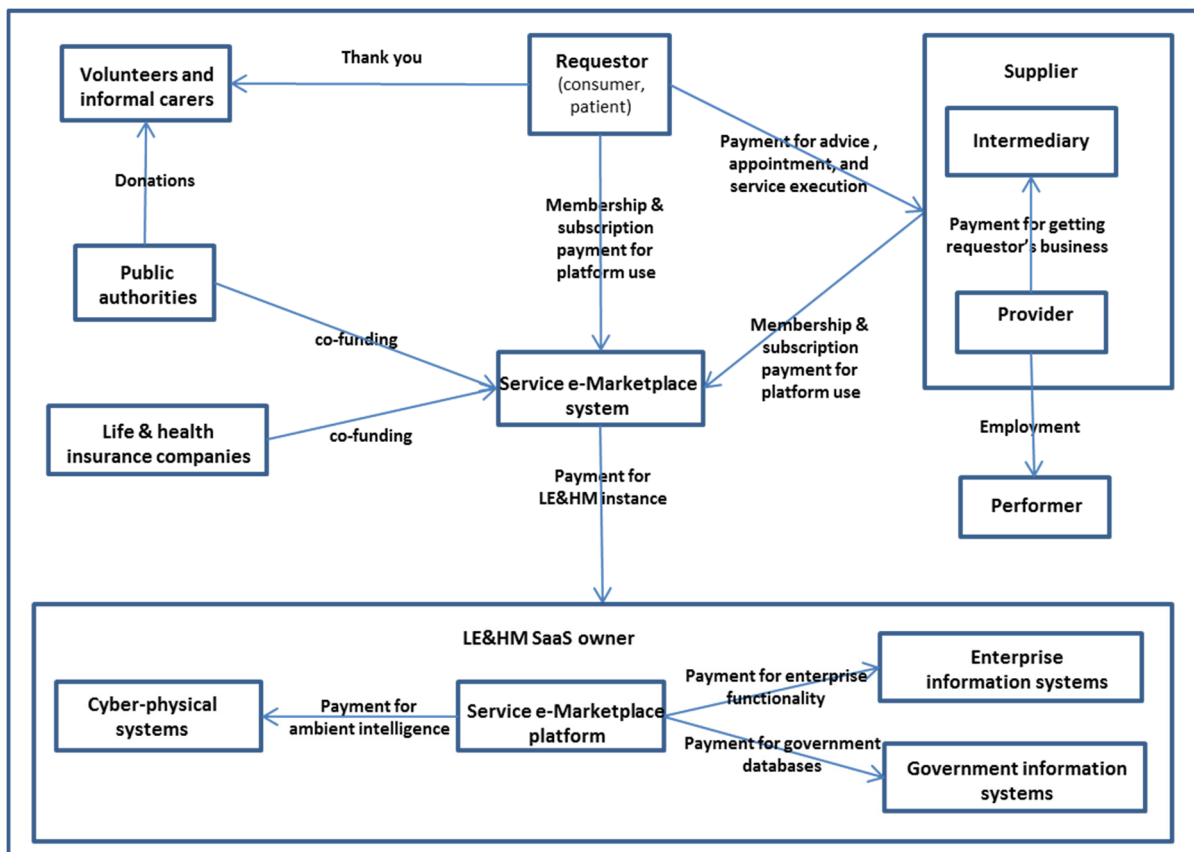


Figure 1: Actors and revenue streams in service e-Marketplace business case.

sensing/computing/actuating capabilities of cyber-physical systems (Haque et al., 2014). Such systems utilize technologies of IoT (Miorandi et al., 2013) and environments of ambient intelligence (Cook et al., 2009).

A troublesome side-effect of any chosen service e-Marketplace business model is the cold start problem. How to entice service providers and performers to pay for and use a vendor's service e-Marketplace platform, if there are no requestors yet? On the other hand, how to entice requestors to request services prior to the establishment of a significant presence of providers and performers on the platform with their service offers? One way of alleviating the cold start problem is by creating local communities of requestors, which not only can create "volume" to attract providers and performers, but can even use barter trade to switch between the roles of requestor and provider/performer.

Technological novelty and attractiveness of a service e-Marketplace platform is a necessary condition for a business success, but it is not a sufficient condition. By observing "rising stars" in service e-Marketplaces, such as airbnb.com, in our R&D we identify and subscribe to five major success factors:

1. Trust (service e-Marketplace platform must provide mechanisms facilitating trust between all actors; the mechanisms must integrate the remaining four factors into a coherent trust-building endeavour).
2. Guarantee (service e-Marketplace platform must offer safeguards to actors in a monetary form for such things as lost business to a performer or damages to a requestor).
3. Payment system (service e-Marketplace payment system must offer a trustworthy and reliable means of transferring money between the actors; the means should include refunds, cancellation policies, protection under Terms of Service, etc.).
4. Verification (service e-Marketplace platform must ensure a proper identification and verification of performers and requestors prior to allowing them to use the system and must seek opinions after a service delivery concluded).
5. Community (most services offered via a concrete instantiation of the service e-Marketplace platform are "localized", i.e. services are enacted in geographically-closed communities; accordingly the platform must facilitate community building by allowing barter services etc. and permitting the community to evolve the platform).

4 SOFTWARE ENGINEERING CHALLENGES

Because the IT is an empowerment and enhancer in the service economy and because older persons are asset and potential engine in the service economy, then there is an obvious need to provide to seniors a friendly easy-to-use service e-Marketplace platforms. This is a huge challenge because of the "lost in the cyberspace" resistance of older persons to using Internet via mobile and non-mobile devices (Heart and Calderon, 2013). A corresponding challenge is that the delivery of software and systems that are easy to use implies that they are difficult to design and implement.

In service e-Marketplaces, the delivery of services to actors is performed (typically) on Everything-as-a-Service models (Banerjee, 2011), in which software, platform and infrastructure are made available as services paid for according to the usage. This creates ubiquitous marketplace where commercial, social, government, health, education and other services are facilitated, negotiated, coordinated and paid for through service e-marketplace platforms.

Figure 2 illustrates a requestor-centric view of service e-Marketplace processes. The diagram distinguishes between the IL and AAL environments and contexts. In both environments, ambient intelligence (Cook et al., 2009; Sadri, 2011) has an important role to play. Immobile (residential, fixed) and mobile ambient intelligence devices (sensors, actuators) cover the spectrum of requestor (consumer, patient) monitoring (daily living, health status, intensive care). They provide input to cyber-physical systems (Hague et al. 2014), which in turn supply information to concierge systems (intermediaries in Figure 1), to EHR systems (Frączkowski et al., 2014), and to emergency and alerts handling facilities. In parallel, requestors provide real-time input to concierge and EHR systems as well as can request emergency assistance.

The upper part of Figure 2 relates to the services required by a requestor. Some services can be automatically or semi-automatically identified through the process of service matching and selection. Although not directly shown in Figure 2, this process can be triggered in a way oblivious to a requestor - it can be cyber-physical or concierge/intermediary initiated. It can also be activated and negotiated directly by a requestor. Service order and execution can flow from service matching and selection. In all cases, a requestor will be directly involved in ordering and execution of a service. The process itself can be advised by experts

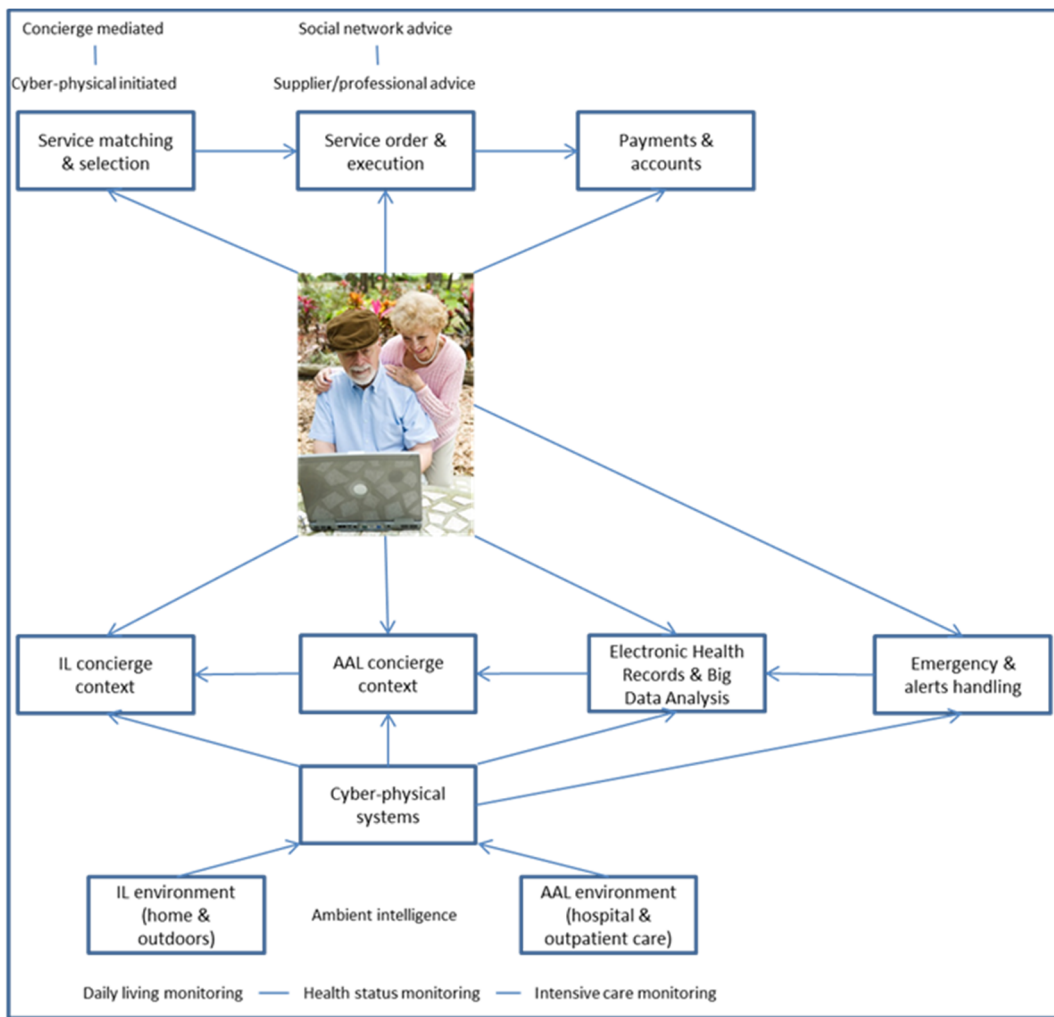


Figure 2: Requestor-centric view of service e-Marketplace processes.

(suppliers and other professionals) and by social networks. Finally, the process of payments and accounts is noted in Figure 2.

The established discipline of software engineering (e.g. Maciaszek and Liang, 2005) advocates architecture-driven forward-engineering processes. Modern practices challenge such processes. They also challenge the rigid top-down development epitomized in three consecutive phases of systems analysis, design and implementation.

While the architectural patterns, principles, and frameworks remain a valid software engineering objective for development of a service e-Marketplace platform (ref. e.g. Maciaszek et al., 2015a; Maciaszek et al., 2015b), the ways of achieving that objective have changed in modern practices. Main practices that underlie this trend are DevOps and microservices.

DevOps is an approach for merging development

and operations (Huttermann, 2012). On the level of user interface and web programming, DevOps is supported by approaches known as responsive development, progressive enhancement, and graceful degradation (Overfield et al., 2013).

The microservice architectural style is “an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API” (Lewis and Fowler, 2014).

Together, DevOps and microservices allow for an agile development while not jeopardizing the most crucial software quality - adaptability (i.e. understandability, maintainability and evolveability). With DevOps plus microservices at work, the owner/vendor of service e-Marketplace platform sets up cloud tenant instances for service e-marketplace businesses. Service e-Marketplace system is then a cloud instance of a generic Software-as-a-Service

platform. Customization and variability of instances is based on the technology of multi-tenancy and uses the emerging principles of Service Line Engineering (SLE) (Mohabbati et al., 2013; Walraven et al., 2014)).

5 CONCLUSIONS

The service e-Marketplace research agenda needs to focus on the dedicated science and engineering for designing software systems and applications supporting provision of e-service solutions in general, and specifically for independent and assisted living of older persons. By centering on consumerization, personalization, and collaborative context-dependent value creation, the R&D into service e-Marketplace shifts towards what Brenner et al., (2014) call user, use & utility research (or 3U research, to use another parlance).

The service e-Marketplace research agenda for lifestyle engineering and health management must conduct both fundamental and pragmatically-minded research along the design science principles (Hevner and March, 2003; Hevner et al., 2004; Niederman and March, 2012, Oesterle et al., 2010). While design science is the preferred scientific method for much of information systems research, it becomes the necessity for R&D into service information systems, as per three convincing arguments identified by Boehmann et al. (2014).

1. Service systems cannot be meaningfully validated in laboratory experiments. An external validity in real-world contexts (piloting of innovation) is necessary.
2. Novel service systems can only emerge with actors' acceptance and involvement (through participatory design and action research).
3. Service systems engineering develops software models and artifacts and is, therefore, design-oriented by their very nature.

Within the realm of design science, the R&D should follow the recommendation and practice of consortium research (Oesterle and Otto, 2010). The consortium research method bridges the practitioners' knowledge with scientific objectives. In effect, it bridges practical relevance with scientific rigor.

Within the realm of consortium research, the service e-Marketplace embraces the innovation chain as pictured in Figure 3. The diagram extends the model of evolution of the engineering discipline proposed by Shaw (2009). The innovation chain originates from a craft practice of available

technology that leads to some localized use of it. If successful, the craft practice evolves into a production line under management control. This results in a commercial market for a line of products. Competition and a natural human inquiry lead to science, based on observations, experiments and proofs in order to confirm stipulated theories. When the volume of confirmed theories gets sufficiently significant, an engineering discipline emerges and allows production with predictable outcomes. Consequently, a related economy emerges as a network of producers, distributors and consumers of products and services based on these products. The final beneficiary of the innovation chain is the society at large.

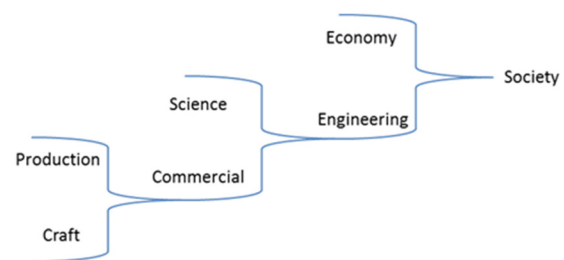


Figure 3: Innovation chain in service e-Marketplace research (extended from Shaw (2009)).

An important underpinning of the service e-Marketplace research is a desire for using rigorous qualitative and quantitative investigation methods. While qualitative and quantitative research can be conducted independently and in parallel, there is a realization that quality characteristics of software systems can and should be quantified, as evidenced in software metrics research (Fenton and Pfleeger, 1997; SQuaRE, 2011).

In a more fundamental sense, any contemporary service e-Marketplace platform for senior persons is a realization of what Gregory Abowd calls the fourth generation of collective computing that integrates the cloud, the crowd, and the shroud (Abowd, 2016). In this collective computing scenario, the cloud delivers endless and always available resources, the crowd contributes collective intelligence of connected humans, and the shroud offers cyber-physical computing of connected immobile and mobile devices. While the cloud and the crowd have broader applicability, the shroud is particularly important in application domains such as personal health and wellness, and therefore in the platforms for independent and assisted living.

ACKNOWLEDGEMENTS

I am grateful to Hubert Oesterle for his constructive comments on earlier versions of this paper and for important hints how to extend and improve it.

DISCLAIMER

A slightly different version of this paper was presented, but not published, at the ASEAN-European Academic University Network, Scientific Meeting, Udayana University, Bali, 15-18 February 2016.

REFERENCES

Abowd, G.D., 2016. Beyond Weiser: From Ubiquitous to Collective Computing, *IEEE Computer*, January, pp.17-23.

Aguter, C., 2013. *ITIL® Lifecycle Essentials. Your essential guide for the ITIL Foundation exam and beyond*, IT Governance Publishing, 375p.

Banerjee, P. et al., 2011. Everything as a Service: Powering the New Information Economy, *Computer (IEEE)*, March, pp.36-43.

Biziel, G. Franczyk, B. & Slowik, J., 2016. Efficiency of the Appointment Coordination Process' Model in the Emerging Business (manuscript accepted for publication at the *Sixth International Conference on Business Intelligence and Technology. BUSTECH 2016*).

Boehmann, T., Leimeister, J.M. and Moeslein, K., 2014. Service Systems Engineering. A Field for Future Information Systems Research, *Business & Information Systems Engineering*, 2, pp.73-79.

Brenner et al., 2014: User, Use & Utility Research. The Digital User as New Design Perspective in Business and Information Systems Engineering, *Business & Information Systems Engineering*, 1, pp.56-61.

Cook, D. J., Augusto, J. C and Jakkula, V. R., 2009. Ambient Intelligence: Technologies, Applications and Opportunities, *Pervasive and Mobile Computing*, 5, pp.277-298.

Cusumano, M. A., 2008. The Changing Software Business: Moving from Products to Services, *IEEE Computer*, January, pp.20-27.

Fenton, N. E. and Pfleeger, S. L., 1997. *Software Metrics. A Rigorous and Practical Approach*, 2nd ed., PWS Publishing Company, 638p.

Frączkowski, K. Mazur, Z. and Mazur, H., 2014. Preconditions for Processing Electronic Medical Records, in: *BDAS 2014*, CCIS 424, S. Kozielski et al. (eds), Springer, pp.504-514.

Haque, S. A., Aziz, S. M. and Rahman, M., 2014. Review of Cyber-Physical System in Healthcare, *International*

Journal of Distributed Sensor Networks, Article ID 217415, 20p.

Heart, T. and Kalderon, E., 2011. Older Adults: Are They Ready to Adopt Health-related ICT?, *International Journal of Medical Informatics*, 82, pp.e209-e231

Hess, T. Legner, C. Esswein, W. Maass, W. Matt, C. Oesterle, H. Schlieter, H. Richter, P and Zarnekow, R., 2014. Digital Life as a Topic of Business and Information Systems Engineering?, *Business & Information Systems Engineering*, Published online: 17 July 2014, 7p.

Hevner, A. R. and March, S. T., 2003. The Information Systems Research Lifecycle, *Computer (IEEE)*, November, pp.111-113.

Hevner, A. R. et al., 2004. Design Science Research in Information Systems, *MIS Quarterly*, 28, 1, pp.75-105

Huttermann, M., 2012. *DevOps for Developers*, Apress.

Laing, A. and Hogg, G., 2002. Political Exhortation, Patient Expectation and Professional Execution: Perspectives on the Consumerization of Health Care, *British Journal of Management*, Vol. 13, pp. 173-188.

Lewis, J. and Fowler, M., 2014. Microservices. A Definition of this New Architectural Term, Retrieved from: <http://martinfowler.com/articles/microservices.html#CharacteristicsOfAMicroserviceArchitecture>

Maciaszek, L. A., Liong, B. L., 2005. Practical Software Engineering. A Case-Study Approach. Addison-Wesley.

Maciaszek, L.A. and Skalniak, T., 2015a. Confluent Factors, Complexity and Resultant Architectures in Modern Software Engineering. A Case of Service Cloud Applications, In: *Fifth International Symposium on Business Modeling and Software Design BMSD 2015*, ed. B. Shiskov, ScitTePress, pp.27-45.

Maciaszek, L.A. Skalniak, T. and Biziel, G., 2015b. Architectural Principles for Service Cloud Applications, In: *Business Modeling and System Design. 4th International Symposium BMSD 2014. Revised Selected Papers*, ed. B. Shiskov, Lecture Notes in Business Information Processing LNBIP 220, Springer, pp.1-21.

Miorandi, D., Sicari, S., Pellegrini De, F. and Chlamtac, I., 2012. Internet of Things: Vision, Applications and Research Challenges, *Ad Hoc Networks*, 10, pp.1497-1516.

Mohabbati, B. Asadi, M. Gasevic, D. Hatala M. and Mueller, H.A., 2013. Combining Service-Oriented and Software Product Line Engineering: A Systematic Mapping Study, *Information and Software Technology*, Article in press <http://dx.doi.org/10.1016/j.infsof.2013.05.006>, 15p.

Niederman, F. and March, S.T., 2012. Design Science and the Accumulation of Knowledge in the Information Systems Discipline, *ACM Transactions on Management Information Systems*, Vol. 3, No. 1, pp.1:3-1:15.

Oesterle, H., 2014. Business oder Life Engineering?, *HMD Praxis der Wirtschaftsinformatik*, Springer, Volume 51, Issue 6, pp 744-761.

Oesterle, H. and Otto, B., 2010. Consortium Research. A

- Method for Researcher-Practitioner Collaboration in Design-Oriented IS Research, *Business & Information Systems Engineering*, 5, pp.283-293.
- Oesterle, H. et al., 2010. Memorandum on Design-Oriented Information Systems Research, *European Journal of Information Systems*, pp.1-4.
- Overfield, E., Zhang, R., Medina, O. and Khipple, K., 2013. Responsive Web Design and Development. In Overfield, E., Zhang, R., Medina, O. and Khipple, K. (Eds), *Pro SharePoint 2013 Branding and Responsive Web Development*, pp.17-46, Apress.
- Osl, P. Sassen, E. and Oesterle, H., 2008. A Guideline for the Design of Collaborative Business Models in the Field of Ambient Assisted Living, *Deutscher AAL-Kongress mit Ausstellung*. Berlin, Offenbach: VDE Verlag GmbH, pp.179-183. http://works.bepress.com/hubert_oesterle/169.
- Pritchard, R. E. and Potter, G. C., 2011. Senior Citizens, Social Security, and Healthcare Costs, *Journal of Business & Economic Research*, Vol. 9, No. 1, pp.23-28.
- Sadri, F., 2011. Ambient Intelligence: A Survey, *ACM Computing Surveys*, Vol. 43, No. 4, Article 36, 66p.
- Shaw, M., 2009. Continuing Prospects for an Engineering Discipline of Software, *IEEE Software*, November/December, pp.64-67.
- SQuaRE, 2011. *Systems and software engineering — Systems and Software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*. ISO/IEC 25010, 44p.
- Targowski, A., 2009. The Architecture of Service Systems as the Framework for the Definition of Service Science Scope, *International Journal of Information Systems in the Service Sector*, 1(1), pp.54-77.
- UN, 2015. World Population Prospects. Key Findings & Advanced Tables. 2015 Revision, United Nations, 66p.
- Walraven, S. et al., 2014. Efficient Customization of Multi-Tenant Software-as-a-Service applications with Service Lines, *The Journal of Systems and Software*, 91, pp.48-62.

DP-CORE: A Design Pattern Detection Tool for Code Reuse

Themistoklis Diamantopoulos, Antonis Noutsos and Andreas Symeonidis

Electrical and Computer Engineering Dept., Aristotle University of Thessaloniki, Thessaloniki, Greece
thdiaman@issel.ee.auth.gr, anoutsos@auth.gr, asymeon@eng.auth.gr

Keywords: Design Pattern Detection, Static Code Analysis, Reverse Engineering, Code Reuse.

Abstract: In order to maintain, extend or reuse software projects one has to primarily understand what a system does and how well it does it. And, while in some cases information on system functionality exists, information covering the non-functional aspects is usually unavailable. Thus, one has to infer such knowledge by extracting design patterns directly from the source code. Several tools have been developed to identify design patterns, however most of them are limited to compilable and in most cases executable code, they rely on complex representations, and do not offer the developer any control over the detected patterns. In this paper we present DP-CORE, a design pattern detection tool that defines a highly descriptive representation to detect known and define custom patterns. DP-CORE is flexible, identifying exact and approximate pattern versions even in non-compilable code. Our analysis indicates that DP-CORE provides an efficient alternative to existing design pattern detection tools.

1 INTRODUCTION

Developers need to understand existing projects in order to maintain, extend, or reuse them. However, understanding usually comes down to understanding the source code of a project, which is inherently difficult, especially when the original software architecture and design information is unavailable. And, although several tools extract information from source code, in cases where software projects lack proper documentation, the process of understanding the intent and design of the source code requires a lot of effort.

The design decisions taken during software development concern the non-functional aspects of the system, and are usually documented in the form of *design patterns*. Design patterns provide reusable solutions in the form of templates that developers can use to confront commonly occurring problems (Gamma et al., 1998). Inferring such non-functional knowledge from source code typically requires extracting these patterns. Lately, the problem of recovering design patterns from source code has attracted the attention of several researchers and has led to the development of several tools to detect patterns, known as *Design Pattern Detection (DPD)* tools.

Most of the tools are effective for detecting certain types of design patterns. However, they fall short in several important aspects. At first, they require the source code to be compilable, or in most cases ex-

ecutable. As a result, developers cannot exploit the source code of other systems without first resolving their dependencies and executing them correctly. Secondly, pattern representations in most tools are not intuitive, thus resulting in black box systems that do not allow the developer any control over the detected patterns. These tools also do not offer the ability to define custom patterns. Finally, several DPD tools are not up-to-date, supporting only obsolete versions of programming languages.

In this paper, we present DP-CORE, a Design Pattern detection tool for COde REUse, which is designed in order to overcome the aforementioned issues. DP-CORE uses a highly descriptive and complete representation for source code elements based on UML. Hence, the tool supports both the detection of several well known patterns and the definition of custom patterns by the developer. DP-CORE is also quite flexible, matching not only strictly defined versions of patterns but also similar versions of patterns using wildcards. Furthermore, the detection of patterns in non-executable and even non-compilable source code is fully supported, while DP-CORE also uses the latest compiler technology to support detecting patterns in current Java projects.

The rest of this paper is organized as follows. Section 2 provides background knowledge and reviews current approaches for the detection of design patterns from source code. In Section 3, we present our DPD

tool, focusing on the defined representation and the methodology used to detect patterns, while its user interface is presented in Section 4. Section 5 illustrates the usage of our tool using a case study and presents our evaluation against the DPD tool PINOT (Shi and Olsson, 2006). Finally, Section 6 summarizes work done and provides useful insight for future research.

2 BACKGROUND AND RELATED WORK

Software design patterns were first proposed in 1987 (Beck and Cunningham, 1987), shortly after the *Object Oriented Programming (OOP)* paradigm began gaining momentum. One of the first attempts to formalize the use of design patterns was a book published in 1998 by Gamma, Vlissides, Johnson, and Helm (1998), who became known as the *Gang of Four (GoF)*. The GoF defined 23 patterns, which are categorized in *creational*, *structural*, and *behavioral* patterns. Creational patterns abstract the process of creating objects, while structural patterns combine objects to form larger structures and behavioral patterns describe the communication between objects.

Current literature in DPD methods is quite broad. According to Dong et al. (2009), the defining properties of a DPD tool are (a) the type of input, which can be source code, UML diagrams or any other representation, (b) the intermediate representation, which includes all structural formats used by the DPD tool, e.g. *Abstract Syntax Tree (AST)*, *Call Dependency Graph (CDG)*, (c) the type of analysis, performed either only on source code or also on the dynamic execution trace, and (d) the type of recognition, which can be either exact or approximate. Although all of the above properties are important, the main classifying feature of DPD techniques is the type of analysis. Following a classification similar to that of current literature (Rasool and Streitferdt, 2011; Dong et al., 2007b), we distinguish among *structural analysis*, *behavioral analysis*, and *semantic analysis* techniques.

Structural analysis approaches detect patterns using information extracted from inter-class relationships (e.g. class inheritance, associations, etc.), thus they are particularly efficient for identifying creational and structural patterns. By contrast, behavioral analysis approaches employ dynamic program analysis, using runtime information to distinguish among structurally identical patterns. Semantic analysis is usually complementary to the structural and behavioral types, aspiring to reduce the false positive rates of structural and behavioral approaches by using semantic information (e.g. naming conventions).

Several DPD tools in current literature employ static code analysis techniques to identify patterns. One of the first tools in this category is the DPD tool by Tsantalis et al. (2006), which extracts static information about the software project including inheritance, method signatures, etc., and represents the project as a graph. Relationships among objects are represented using matrices, while the tool uses similarity algorithms to detect patterns. A slightly different approach is followed by Lucia et al. (2009) for the DPRE tool, which is implemented as an Eclipse plugin. The tool constructs UML diagrams and uses visual language parsing techniques to identify patterns.

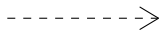

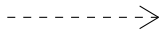
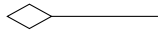
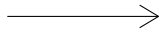

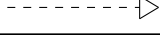
PINOT, designed by Shi and Olsson (2006), is another popular tool that combines both structural and behavioral analysis. It extracts information from the AST of the source code, and detects patterns using structural and behavioral (data flow) template matching. Several tools also use machine learning methods. Arcelli and Christina (2007) developed MARPLE, an Eclipse plugin that uses neural networks to classify source code representations to behavioral patterns. The authors also extended their work, introducing JADEPT (Arcelli et al., 2008), a tool that represents patterns as combinations of rules.

FUJABA (Nickel et al., 2000) is another Eclipse plugin, which, among other functions, supports detecting design patterns. The tool defines patterns using UML class diagrams and expresses their behavioral aspects using story-diagrams, i.e. a combination of activity and interaction diagrams. Metamodel-based approaches are also popular. Guéhéneuc and Antoniol (2008) introduced DeMIMA, which extracts classes, methods, etc. from source code to instantiate a metamodel that is used to specify objects and their relationships. The tool employs constraint programming techniques to identify patterns. A similar approach is followed by Ptidej, designed by Kaczor et al. (2006), which uses a constraints solver to detect sets of objects that are similar to design patterns.

Another notable tool is D-CUBED, by Stencil and Wegrzynowicz (2008), which formulates design patterns in first order logic and inserts source code into a database where patterns can be identified via queries. Finally, DP-Miner by Dong et al. (2007a) is another quite interesting tool that also uses semantics to distinguish among certain patterns that may have similar structural and behavioral aspects. The tool represents source code entities and relationships in XMI and analyzes class and method names to identify patterns.

The tools analyzed in the previous paragraphs are quite effective for detecting different types of patterns. However, their applicability is limited to compilable and executable projects. In specific, tools us-

Table 1: DP-CORE Connections.

Connection Type	Description	UML Relation	UML Symbol
A calls B	A method of class A calls a method of class B	Dependency	
A creates B	Class A creates an object of type class B	Composition	
A uses B	A method of class A returns an object of type B	Dependency/ Multiplicity	
A has B	Class A has one or more objects of type B	Aggregation	
A references B	A method of class A has as parameter an object of type B	Association	
A inherits B	Class A inherits or implements class B or class A realizes interface B	Inheritance/ Realization	 

ing dynamic analysis techniques, such as D-CUBED [17] or DP-Miner [5], require the code to be executable. Moreover, static analysis tools, such as PINOT [16] or DPD [18], also operate on .class files, therefore require that the developer first resolves any dependencies to compile the examined project. Furthermore, tools that rely on compilers, such as PINOT [16], cannot always compile all projects not only because of dependencies but only because of obsolete compiler versions. Finally, several tools such as DeMIMA [9] or Ptidej [10], use complex representations that do not allow the developer any control over the detected patterns. As a result, these tools do not offer the ability to define custom patterns or the ability to detect incomplete or similar patterns. To the best of our knowledge, no tool presented in this Section and no other tool offers all of the aforementioned features. In the following Section, we present an up-to-date customizable DPD tool that effectively abstracts the methodology of pattern detection even in uncompileable or incomplete source code.

3 DP-CORE: A DESIGN PATTERN DETECTION TOOL

In this Section, we describe our DPD tool, DP-CORE, in detail. Subsection 3.1 describes the representation used for source code objects and relationships, while subsection 3.2 provides details about the representation of design patterns by our system. The extraction of entities and relationships from source code is presented in subsection 3.3 and the algorithm for detecting patterns is described in subsection 3.4. Subsection 3.5 illustrates how the results are grouped in order to form combined patterns.

3.1 Representing Objects and Relationships

As already mentioned, an effective structural representation for source code and design patterns should be complete as well as intuitive. Given that most developers are more or less familiar with UML, the representation of DP-CORE is related to UML entities and relationships. Given intuition from (Birkner, 2007), we define two concepts for our representation: the abstraction of each class and the connection between two classes. The abstraction types that we defined for each class, including their description, are shown in Table 2.

Table 2: DP-CORE Abstraction Types.

Abstraction Type	Description
Normal	a non-abstracted class
Abstract	a Java abstract class
Interface	a Java interface
Abstracted	an abstract class or an interface
Any	any of the above class types

The type `Normal` refers to a simple non-abstracted class, while types `Abstract` and `Interface` correspond to the known Java abstract classes and interfaces. Additionally, we define the type `Abstracted` as either one of types `Abstract` and `Interface`, while the type `Any` denotes any of the above abstraction types and functions as a wildcard. Apart from their abstraction types, the classes of the examined source code connect to each other with directional relationships. We define 6 types of connections that are summarized in Table 1, including their description, the corresponding UML relation and the corresponding UML symbol for each connection.

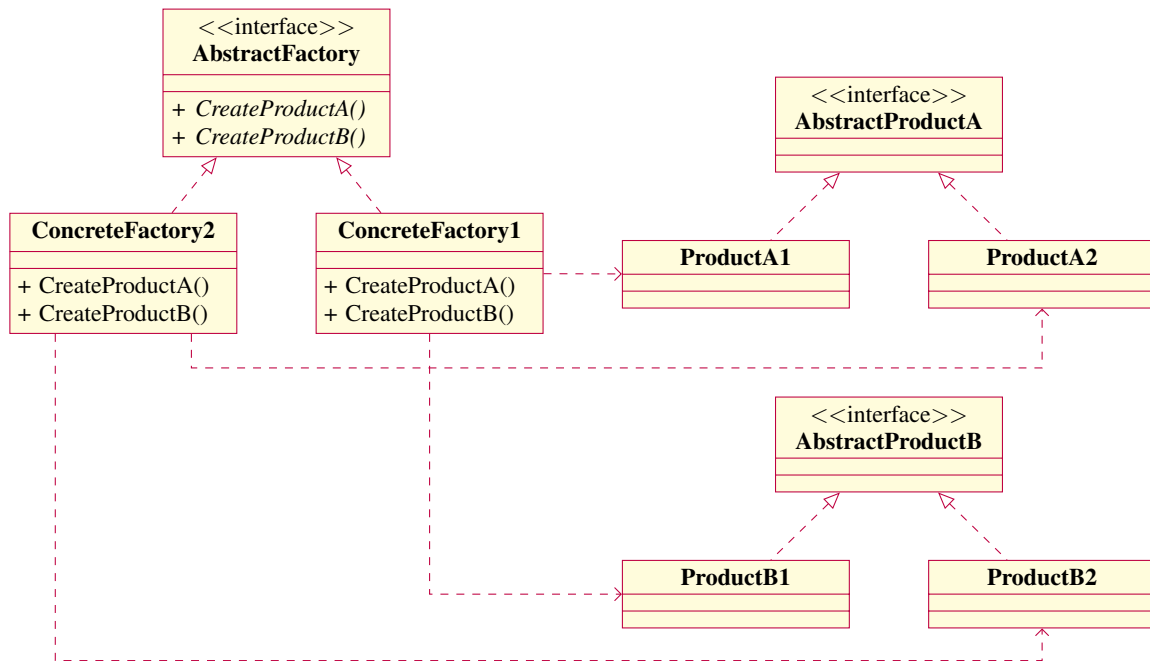


Figure 1: Abstract Factory Pattern.

The connections cover all possible relations that can exist in a source code project. UML dependencies and associations are handled by connections `calls/uses` and `references` respectively, while compositions and aggregations correspond to the `creates` and `has` connections. Inheritance and realization relations are handled by the `inherits` connection. Finally, we define a `relates` connection that covers all possible connection types, and thus is used as a wildcard.

3.2 Representing Design Patterns

Upon having presented how source code entities and relationships are represented in our system, we illustrate how well known (or custom) design patterns can be represented by our system. For each pattern, one must define the abstraction of its member classes and the connections among them. In this subsection, we illustrate how the Abstract Factory pattern is defined. According to the GoF (Gamma et al., 1998), the purpose of the Abstract Factory pattern is to “provide an interface for creating families of related or dependent objects without specifying their concrete classes”.

Figure 1 depicts the interface `AbstractFactory` which defines the methods that are implemented by `AbstractProductA` and `AbstractProductB`, while `ConcreteFactory1` and `ConcreteFactory2` define the methods of the `Product` objects. Note that this is one possible illustration, since it is possible to have different number of `ConcreteFactory`, `AbstractProduct` and/or `Product` classes. By def-

inition, an Abstract Factory pattern has to include instances of `AbstractFactory`, `AbstractProduct`, `ConcreteFactory`, and `Product`. Using the representation of subsection 3.1, we define the 4 members of the pattern and their connections in Figure 2.

```

Abstract Factory
A Normal ConcreteFactory
B Abstracted AbstractFactory
C Normal Product
D Abstracted AbstractProduct
End_Members
A inherits B
C inherits D
A creates C
A uses D
End_Connections
    
```

Figure 2: Abstract Factory Pattern File.

For each pattern member, A, B, C, and D, we can see its abstraction type and its ability. The ability of a member is a description field that can be set by the developer so that the detected pattern instances are more comprehensive. The pattern has 4 connections, 2 inherits, one from A (`Concrete Factory`) to B (`Abstract Factory`) and one from C (`Product`) to D (`Abstract Product`), 1 creates from A (`Concrete Factory`) to C (`Product`), and 1 uses from A (`Concrete Factory`) to D (`Abstract Product`).

```

public class Car extends Vehicle{----- Car inherits Vehicle

    private Model model;----- Car has Model
    private Fuel fuel;----- Car has Fuel

    public Car (Model model,----- Car references Model
                float fuelCapacity){
        this.model = model;
        fuel = new Fuel(fuelCapacity);----- Car creates Fuel
    }

    public Model getModel(){----- Car uses Model
        return model;
    }

    public void addFuel(float fuelQuantity){
        fuel.add(fuelQuantity);----- Car calls Fuel
    }
}

```

Figure 3: Example of Extracting Connections for a Car Class.

3.3 Extracting Objects and Relationships from Source Code

The main building blocks of our methodology are objects and relationships between them. DP-CORE uses the Java Compiler Tree API (Oracle, 2015) to extract these elements. Upon extracting the AST for each file, all class objects are extracted from the code, including their abstraction type. After that, DP-CORE extracts the connections. A uses connection from A to B is extracted if the return type of any method of A is an object of type B. Connections of type inherits are extracted if class A extends or implements class B. For the has connection from A to B, all variables of A are checked to find if any of them is of type B. For the calls connection, all method invocations of class A are checked whether the method invoked is a method of class B. Connections of type creates are extracted if class A constructs a new object of type B. Finally, a references connection from A to B is declared if any method of A has as parameter an object of type B.

Figure 3 depicts an example of extracting the connections of a Car class which interacts with three classes. It inherits the Vehicle class and has two objects of type Model and Fuel. Additionally, Car references the Model in its constructor, where the Fuel object is also created. Finally, the getter function of Car also implies that it uses the Model class, while Car also calls a method of Fuel to add fuel to its tank. Hence, we can define 7 connections among the classes in this example, which are shown in the annotations on the right of Figure 3.

3.4 Design Pattern Detection Algorithm

Upon having extracted the objects and relationships of the examined software project, the next step is to detect patterns in the extracted structure. Although this problem could be solved by iterating over all possible permutations of classes, this brute force method would be computationally inefficient. For instance, given a project with 50 classes and a pattern with 4 members, this method would check more than 5 million permutations. Thus, we designed an algorithm that prunes permutations as it recursively finds pattern candidates.

Input: Objects, Members

Output: Candidates

```

Detect(Objects, Members, Candidate, d):
    if d < Members.length():
        Member = Members[d]
        for Object in Objects:
            if abstraction(Object, Member)
            and connections(Object, Member):
                Detect(Objects \ {Object},
                    Candidate ∪ {Object},
                    Candidate, d + 1)
    else:
        Add Candidate to Candidates

```

Figure 4: Design Pattern Detection Algorithm.

Our algorithm is shown in Figure 4. It receives as input the objects extracted from the examined project (and their connections), as well as the pattern to be

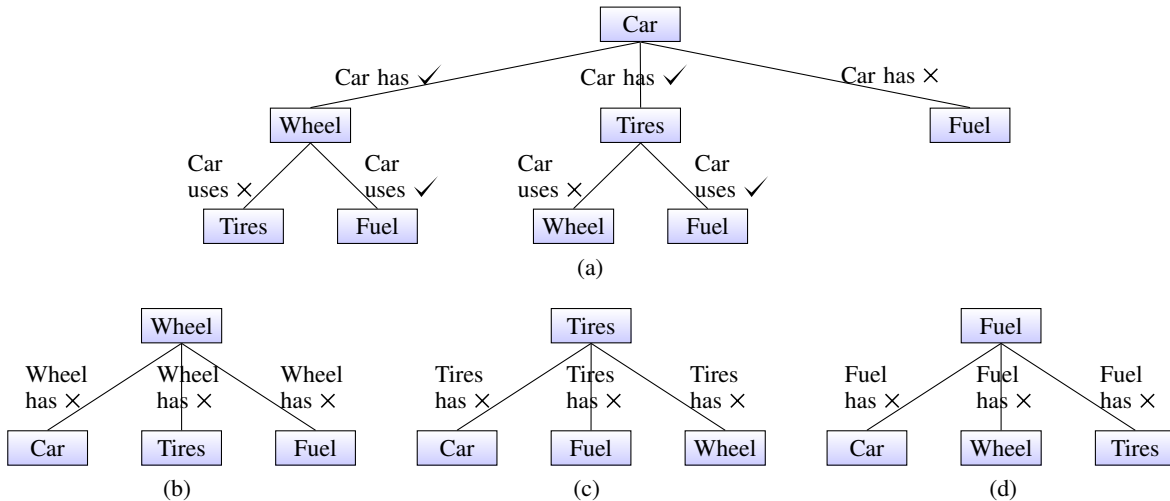


Figure 5: Combinations of objects and their relationships, where the object that matches the first pattern member is (a) Car, (b) Wheel, (c) Tires, and (d) Fuel.

detected in the format defined in the previous subsections. It iterates over the Objects and checks whether the current Object can be matched to the current pattern member. This is performed by recursively calling the Detect function and providing the index to current pattern member as the depth parameter d . At first, the algorithm is initialized with depth equal to 0 (and Candidate is the empty set). Iterating over the first Object, it is checked whether its abstraction and its connections are the same with pattern member 0. If the Object matches this pattern member, then the Detect function is called again given as parameters the Objects without the already matched Object and the updated Candidate so that it includes the Object, while the depth parameter is also incremented. If at any time the current Object does not match the current pattern member, then the recursion stops. When all pattern members are matched, then the Candidate is added to the detected pattern Candidates.

We illustrate the execution of the algorithm using an example with 4 classes Car, Tires, Wheel, and Fuel, connected with the relationships Car has Tires, Car has Wheel, and Car uses Fuel. The pattern to be detected has 3 classes A, B, and C, connected with the relationships A has B and A uses C. To simplify our example let us assume that the objects of the examined code and the members of the pattern do not have any abstraction. Some possible combinations for this example are shown in Figure 5.

Note that given 4 objects, their possible permutations per 3 pattern members are 24. Using our algorithm, however, we examine much fewer. At first, all permutations involving Wheel, Tires, or Fuel in the position of member A (shown in Figures 5(b), 5(c), and 5(d) respectively) are discarded in the first level

of the tree, since they do not connect to any other element with the required connections. If Car is matched with pattern member A, then the other objects are checked for has and uses connections. As shown in Figure 5(a), the path of the connection Car has Fuel is pruned. The other two paths, including the connections Car has Wheel and Car has Tires, are further examined, to finally provide the two instances of the pattern $\{A = \text{Car}, B = \text{Wheel}, C = \text{Fuel}\}$ and $\{A = \text{Car}, B = \text{Tires}, C = \text{Fuel}\}$ respectively.

3.5 Grouping Design Pattern Instances

In a typical design pattern detection scenario, the examined code may have several instances of a design pattern. Since some of these instances may be part of the same design decision, we need a way to merge these candidate patterns to form a unified design pattern. For instance, consider the 4 Abstract Factory candidate instances of Table 3.

DP-CORE merges these instances in two steps. At first, the tool iterates over all candidate patterns and checks whether any of them have identical members except for one. In this case, two super-patterns would be formed, one for the ReptileFactory, i.e. $\{A = \text{ReptileFactory}, B = \text{SpeciesFactory}, C = \text{Snake|Tyrannosaurus}, D = \text{Animal}\}$, and one for the MammalFactory, i.e. $\{A = \text{MammalFactory}, B = \text{SpeciesFactory}, C = \text{Cat|Dog}, D = \text{Animal}\}$. After that, DP-CORE iterates over the super-patterns to identify again the ones having identical members except for one. Thus, the final hyper-pattern for this example is visualized in Figure 6.

Table 3: Abstract Factory Candidate Instances.

Member	Ability	Candidate 1	Candidate 2	Candidate 3	Candidate 4
A	ConcreteFactory	ReptileFactory	ReptileFactory	MammalFactory	MammalFactory
B	AbstractFactory	SpeciesFactory	SpeciesFactory	SpeciesFactory	SpeciesFactory
C	Product	Snake	Tyrannosaurus	Cat	Dog
D	AbstractProduct	Animal	Animal	Animal	Animal

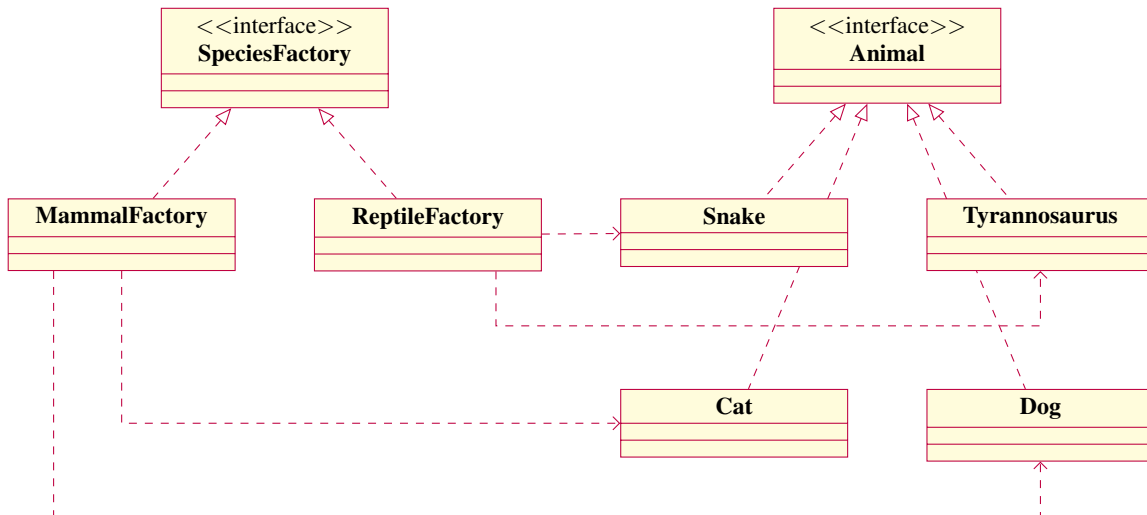


Figure 6: Hyper-Pattern for the Pattern Candidate Instances of Table 3.

4 INTERFACES OF DP-CORE

DP-CORE is implemented in Java, and offers two user interfaces: a *Graphical User Interface (GUI)* and a *Command Line Interface (CLI)*. The source code and the releases are available in the repository:

<https://github.com/AuthEceSoftEng/DP-CORE>

DP-CORE receives as input project folders including .java files for pattern detection, as well as .pattern files that define the pattern to be detected. Its output is a txt file that includes the results of the detection. An example output of DP-CORE is shown in Figure 7.

In this case, the examined project has two candidates of the Command pattern. Since they refer to the same pattern, the user could also merge them by enabling the grouping mechanism of subsection 3.5.

The GUI of DP-CORE is shown in Figure 8. The main screen of the tool, shown in Figure 8(a), includes 3 textfields, which are used to enter the folder where patterns are stored, the folder of the examined project, and the folder where the results are saved. Upon entering this information, the user can select the pattern to be detected using the drop-down menu and push the “Detect Pattern” button. The “Grouping” checkbox controls whether the pattern grouping mechanism of subsection 3.5 is activated. The user can either select

```

Amount of Candidates found: 2

Candidate of Pattern Command:
A(ConcreteCommand): LightOffCommand
B(Command): Command
C(Receiver): Light
D(Invoker): RemoteControl

Candidate of Pattern Command:
A(ConcreteCommand): LightOnCommand
B(Command): Command
C(Receiver): Light
D(Invoker): RemoteControl
    
```

Figure 7: Example Output of DP-CORE.

a predefined pattern or define a new pattern by pushing the “Create Custom Pattern” button. Upon pressing this button, the user is presented with the screen shown in Figure 8(b), where the user names the newly defined pattern and defines the number of its members and connections. After that, the members and the connections of the pattern are defined in two subsequent screens shown in Figures 8(c) and 8(d).

Finally, the tool can be used from the command line to allow the execution of batch jobs with multiple projects and/or patterns. Similarly to the GUI, the CLI of DP-CORE receives as input the project folder

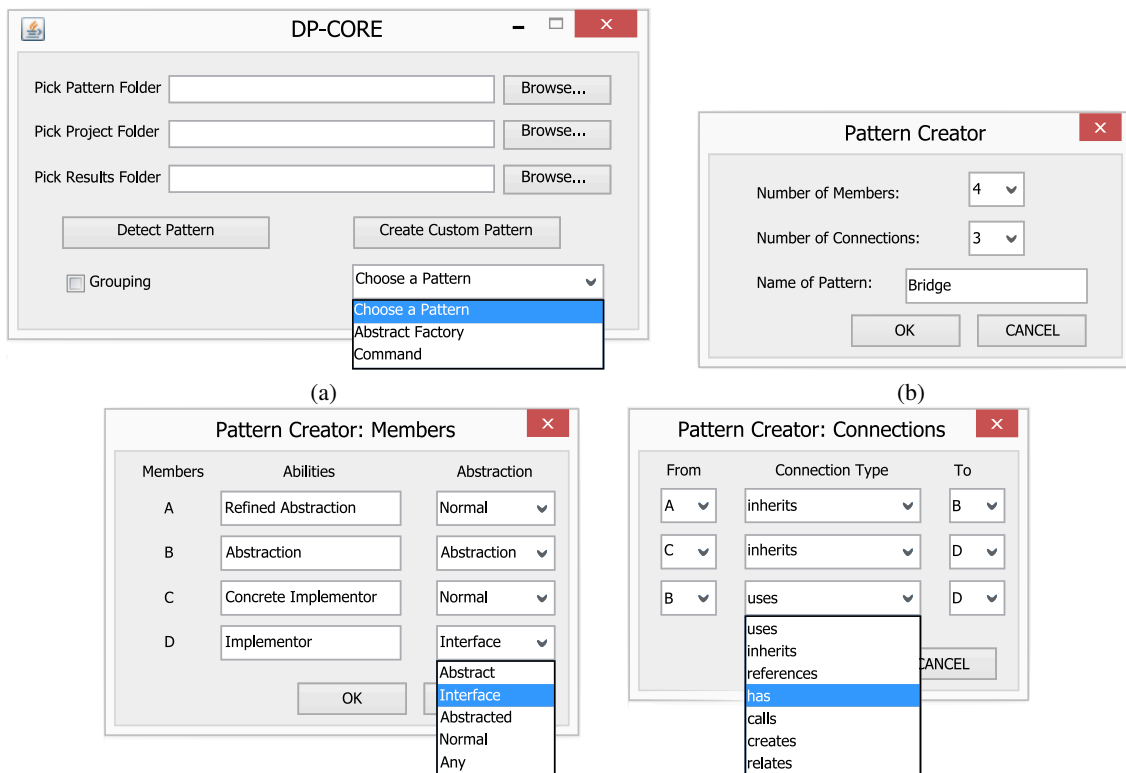


Figure 8: Screenshots of (a) the Main Screen of DP-CORE, (b) the Pattern Creator, including (c) the Members and (d) the Connections Screens of DP-CORE.

as well as a .pattern file containing the pattern to be detected. The output is printed in the console.

5 EVALUATION

We assess the effectiveness of DP-CORE using two evaluation experiments. The first experiment involves an example project including known instances of patterns, while the second experiment involves a comparison to PINOT (Shi and Olsson, 2006) for detecting patterns in the source code of known Java libraries.

5.1 Example Design Patterns Project

For the first experiment we used several design pattern examples merged together in a common project. Although this project is not a typical Java application, it provides an interesting case study for our tool. We created pattern files for 6 GoF patterns of all types: the creational patterns Abstract Factory and Builder, the structural pattern Bridge, and the behavioral patterns Command, Observer and Visitor. The example project and the patterns are provided with our tool as examples. The results are shown in Table 4.

DP-CORE successfully identified all the pattern instances in the project. It is notable, though, that the tool detected false positive instances, since 27.27% of the detected instances are not design patterns. These false positives, however, are due to the non-strict definition of the patterns. For instance, the 4 falsely identified instances of the Command pattern are instances of the Builder pattern, which is expected since the definitions of these two patterns are similar. This is also the case for the Observer pattern which is similar to the Visitor pattern. In any case, DP-CORE is quite effective for identifying patterns, as it detects all instances, while false positives can be minimized by providing more precise definitions of patterns.

Additionally, upon tweaking the code of this project, we conclude that DP-CORE can detect patterns even in non-compilable code. In specific, our tool is not affected by common syntax errors, such as missing brackets, missing semicolons etc. Furthermore, most semantic errors, such as missing imports, missing variable declarations, etc., are safely ignored without influencing the detection of patterns. The errors that affect our tool are purely lexical, e.g. writing `fo` instead of `for` or having a space in a variable name. As a result, DP-CORE supports code reuse scenarios,

Table 4: Pattern Detection Results for Example Project.

Design Pattern	#Instances	#Correctly Detected Instances	#Incorrectly Detected Instances	%Correctly Detected Instances	%Incorrectly Detected Instances
Abstract Factory	4	4	0	100%	0%
Command	2	2	4	100%	66.67%
Bridge	4	4	0	100%	0%
Builder	2	2	0	100%	0%
Visitor	2	2	0	100%	0%
Observer	2	2	2	100%	50%
Total	16	16	6	100%	27.27%

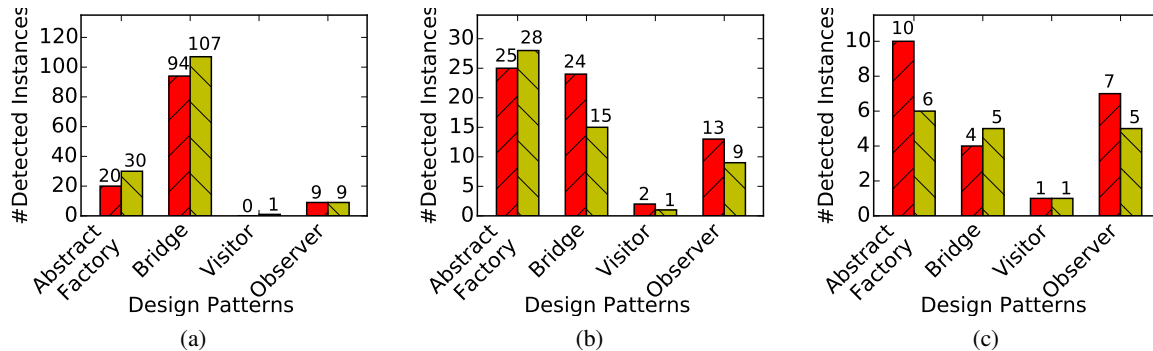


Figure 9: Diagrams of Detected Design Pattern Instances by DP-CORE (▨) and PINOT (▧) for (a) JHotDraw, (b) Java AWT, and (c) Apache Ant.

given that reusable code is usually lexically correct, while it may have omissions in syntax or semantics.

5.2 Java Libraries with Design Patterns

For the second experiment we evaluated DP-CORE against PINOT (Shi and Olsson, 2006) in a dataset of 3 libraries: JHotDraw 6.0b1, Java AWT 1.3, and Apache Ant 1.6.2. PINOT was selected as it resembles DP-CORE in the way that it extracts source code objects and detects patterns. Since, however, the representation used by PINOT differs from that of DP-CORE, the comparison is quantitative, thus the results can only provide a proof-of-concept for our DPD. As already noted, the scope of DP-CORE lies in providing an intuitive way of detecting custom patterns and software architectures, instead of strictly defining the known design patterns. For compatibility reasons, the comparison between the two tools includes the patterns Abstract Factory, Bridge, Visitor, and Observer.

The results for the two tools are shown in Figure 9. For JHotDraw, in Figure 9(a), there are small deviations between the detected patterns of the two tools, which are mostly due to the pattern representation of each tool. Upon manually examining the detected patterns, we conclude that both tools successfully identify the design patterns of this library. The results for the other two libraries, Java AWT in Figure 9(b)

and ApacheAnt in Figure 9(c), are also similar. Note, however, that patterns are not always defined consistently in all projects. As a result, for Java AWT, we had to modify the representation of Abstract Factory, changing the connection `A uses D` to `A uses C` in Figure 2, thus allowing the `ConcreteFactory` to use `Product` instead of `AbstractProduct`. Apache Ant also required a modification to the Bridge pattern, where the `A calls D` connection was added (see Figure 8) to restrict the instances to the ones where the `RefinedAbstraction` calls the `Implementor`.

The main deviations are observed for the Abstract Factory and Bridge patterns, which is expected since the representations of these patterns are similar. Summarizing our analysis, the results of the quantitative comparison between DP-CORE and PINOT indicate that both tools can successfully detect patterns from source code. However, it is important to note that all tools are bound to the representations used to identify patterns, therefore comparing them is not trivial. DP-CORE further allows defining custom patterns and effectively recovers structural design semantics even for non-standard architectures. Finally, concerning execution time, DP-CORE is quite efficient; pattern detection in any project required less than 5 seconds.

6 CONCLUSION

Recovering non-functional design information from source code is a difficult task. Since this information is usually provided in the form of design patterns, several DPD tools have been developed to extract it. However, most tools are limited to identifying certain known patterns in executable projects. In this work, we presented DP-CORE, a DPD tool that can recover patterns even from non-compileable source code. DP-CORE uses a flexible and intuitive representation, allowing developers to define their own patterns and even use wildcards to express ambiguity in these definitions. Our tool is up-to-date, relying on the latest compiler technology, while it offers a GUI and a CLI so that it can be used for batch tasks. The evaluation of DP-CORE has shown that it can be effective for identifying patterns in source code that match the representation provided by the developer.

Future work on our tool lies in several directions. At first, the negation of current connections could be added to the options of the representation, thus allowing the developer to define patterns more strictly. Additionally, the structural analysis of DP-CORE can be further extended using semantics on the abilities of the patterns. Finally, forthcoming versions may include graphical pattern representations by integrating with known graphical editors. In any case, pattern recovery from source code should be connected to the requirements of the developer, thus we believe DP-CORE is an efficient alternative to existing tools.

ACKNOWLEDGEMENTS

Parts of this work have been supported by the FP7 Collaborative Project S-CASE (Grant Agreement No 610717), funded by the European Commission.

REFERENCES

- Arcelli, F. and Christina, L. (2007). Enhancing Software Evolution through Design Pattern Detection. In *Proceedings of the 2007 Third International IEEE Workshop on Software Evolvability*, pages 7–14, Paris, France.
- Arcelli, F. F., Perin, F., Raibulet, C., and Ravani, S. (2008). Behavioral Design Pattern Detection through Dynamic Analysis. In *Proceedings of the 4th International Workshop on Program Comprehension through Dynamic Analysis*, pages 11–16, Antwerp, Belgium.
- Beck, K. and Cunningham, W. (1987). Using Pattern Languages for Object-Oriented Programs. In *Proceedings of the OOPSLA-87 Workshop on the Specification and*
- Design for Object-Oriented Programming*, Orlando, FL, USA.
- Birkner, M. (2007). Object-Oriented Design Pattern Detection Using Static and Dynamic Analysis of Java Software. Master's thesis, University of Applied Sciences Bonn-Rhein-Sieg Sankt Augustin, Germany.
- Dong, J., Lad, D. S., and Zhao, Y. (2007a). DP-Miner: Design Pattern Discovery Using Matrix. In *Proceedings of the 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, ECBS '07*, pages 371–380, Tucson, AZ, USA.
- Dong, J., Zhao, Y., and Peng, T. (2007b). Architecture and Design Pattern Discovery Techniques - A Review. In *Proceedings of the 2007 International Conference on Software Engineering Research & Practice*, volume 2 of *SERP 2007*, pages 621–627, Las Vegas, NV, USA.
- Dong, J., Zhao, Y., and Peng, T. (2009). A Review of Design Pattern Mining Techniques. *International Journal of Software Engineering and Knowledge Engineering*, 19(06):823–855.
- Gamma, E., Vlissides, J., Johnson, R., and Helm, R. (1998). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- Guéhéneuc, Y.-G. and Antoniol, G. (2008). DeMIMA: A Multilayered Approach for Design Pattern Identification. *IEEE Trans. Softw. Eng.*, 34(5):667–684.
- Kaczor, O., Guéhéneuc, Y.-G., and Hamel, S. (2006). Efficient Identification of Design Patterns with Bit-vector Algorithm. In *Proceedings of the 10th European Conference on Software Maintenance and Reengineering, CSMR 2006*, pages 175–184, Bari, Italy.
- Lucia, A. D., Deufemia, V., Gravino, C., and Risi, M. (2009). Design Pattern Recovery Through Visual Language Parsing and Source Code Analysis. *J. Syst. Softw.*, 82(7):1177–1193.
- Nickel, U., Niere, J., and Zündorf, A. (2000). The FUJABA Environment. In *Proceedings of the 22nd International Conference on Software Engineering, ICSE '00*, pages 742–745, Limerick, Ireland.
- Oracle (2015). Compiler Tree API. Avail. online: <http://docs.oracle.com/javase/8/docs/jdk/api/javac/tree/index.html>, [retrieved March, 2015].
- Rasool, G. and Streitferdt, D. (2011). A Survey on Design Pattern Recovery Techniques. *International Journal of Computer Science Issues*, 8(6):251–260.
- Shi, N. and Olsson, R. A. (2006). Reverse Engineering of Design Patterns from Java Source Code. In *Proceedings of the 21st IEEE/ACM International Conference on Automated Software Engineering, ASE '06*, pages 123–134, Tokyo, Japan.
- Stencel, K. and Wegrzynowicz, P. (2008). Detection of Diverse Design Pattern Variants. In *Proceedings of the 2008 15th Asia-Pacific Software Engineering Conference, APSEC '08*, pages 25–32, Beijing, China.
- Tsantalis, N., Chatzigeorgiou, A., Stephanides, G., and Halkidis, S. (2006). Design Pattern Detection using Similarity Scoring. *IEEE Trans. Softw. Eng.*, 32(11):896–909.

SHORT PAPERS

Rule-based Business Process Abstraction Framework

Christina Tsagkani and Aphrodite Tsalgaidou

*Dept. of Informatics & Telecommunications, National & Kapodistrian University of Athens (NKUA),
Panepistimioupolis, Ilisia 157 84, Greece
{tsagkani, atsalga}@di.uoa.gr*

Keywords: Business Process Modeling, Business Process Model Abstraction, BPMN 2.0.

Abstract: Business process models have become key artifacts to represent how work is performed in organisations. Therefore organisations maintain large process model repositories, containing complex models that are difficult to be managed. Abstraction is a means to reduce the size and complexity of such process models and ease the process model management. It is worthwhile noticing that several abstraction mechanisms exist in the literature but none of them includes activities-paths, data, roles, artifacts and messages in the abstraction procedure. To this end we present a conceptual framework that utilizes abstraction rules in order to simplify business process models. The proposed framework is well suited for complex BPMN process models and is built on our previous work where process abstraction rules had been introduced. This abstraction framework is designed with focus on retaining the overall structure of the process model, but most of all it is designed to treat different process elements (eg. data, messages, roles, etc.) other than activities as abstraction objects. A real business case that is originated from the financial services sector is used as an attempt to exemplify and validate the proposed mechanism.

1 INTRODUCTION

Nowadays business process management has attracted the attention by organisations and enterprises which focus on modelling and automating their business processes. Business processes can be a great source of business knowledge. Therefore, a lot of efforts are taking place by organisations in order to build process repositories containing hundreds or even thousands of business process models on behalf of different stake-holders. The way that a process model is defined is influenced by many factors, such as the modeler's expertise (e.g. software engineers, business analysts, domain experts) and the goal of modelling (e.g. models created to provide an overall view of process for managerial use, models defined in a detailed way in order to be executed by technical specialists).

After analysing diverse process models we identified that they exhibit a number of differences:

- Differences in the way that process models achieve the same effect or represent equivalent unit of work
- Differences in the granularity level used to represent the same unit of work
- Differences related to the role assigned to perform

an activity

- Differences in the control-flow relations amongst activities between the process models

Both the heterogeneity that characterises business process models as it is described above and the existence of large-complex repositories, create difficulties in process model management.

Therefore we claim that such differences can be diminished and process management, especially querying, can be improved, if process models are transformed to more coarse-grained models. In this context, abstraction mechanisms are needed that take into consideration such differences and reduce the size and complexity of process models by focusing on the nature of their process elements.

Thus in this paper, we propose a conceptual framework of a rule-based abstraction mechanism that aims at providing a less complex view of business processes by focusing on significant process elements and discarding or aggregating internal process orchestration details by applying specific rules to activities-paths, data objects, messages, roles and artifacts. This mechanism focuses on business processes that are defined using the Business Process Model and Notation (BPMN) (OMG, B.P.M., 2011) as it is a well-known and widely used standard across

different industries. Therefore, the rest of the paper has been structured as follows: section 2 introduces some concepts related to the abstraction mechanism, section 3 describes the proposed abstraction framework, section 4 exemplifies and validates the proposed abstraction mechanism via a real business case, section 5 discusses related process abstraction mechanisms and section 6 concludes and presents our future work.

2 BACKGROUND

2.1 Business Process Modelling

Numerous notations have been emerged into the business process modelling space, including UML Activity Diagrams, the Business Process Modelling Notation (BPMN), Event-driven Process Chains (EPCs), Workflow nets, and the Business Process Execution Language (BPEL) that is more appropriate for executable specifications rather than modelling per se. From all these modelling notations, BPMN 2.0 is the prevailing standard, as it has been widely accepted in industrial practice and we utilise it in our abstraction mechanism.

BPMN aims at documenting and communicating business processes between all business stakeholders. Specifically, it is a graph-based notation — i.e. sets of graphical symbols and rules for combining them — for documenting flow objects, data, connecting objects, swimlanes and artifacts. Flow objects (Events, Activities and Gateways) define the behaviour of business process. Data (Objects, Inputs, Outputs and Stores) define what activities require to be performed or produce. Connecting Objects (Sequence, Message, Associations and Data Associations) define the way the flow objects are connected. Swimlanes (Pools and Lanes) represent process participants. Artifacts (Group and Text Annotation) are used to provide additional information about the process. As defined by (Chinosi and Trombetta, 2012): Process Orchestration includes the private and public processes of an organization. Private Processes are processes internal to a specific organization whereas Public processes represent the interaction between a private process and another process or participant that means only activities that are used to communicate with other participants are included in the public process. On the other hand Choreographies define the expected behaviour that is a procedural contract between interacting participants. Therefore a choreography exists between pools (or participants)

as it bisects the message flows amongst them.

The current framework is focused on obtaining a process quick view by preserving the overall process structure. Therefore we suggest that the abstraction mechanism leaves intact process elements that constitute process's choreography and abstracts only process's orchestration details.

2.2 Process Abstraction

Process Abstraction is a means of providing different process views (which retain information relevant for a particular purpose) and reducing the size and complexity of process models by preserving essential properties and leaving out insignificant details (Smirnov et al., 2010). It may be applied for different purposes such as focus on specific process model properties (i.e. preserve pricey/frequent/long activities), adapt process model for an external partner, trace data/task dependencies and obtaining a process quick view respecting ordering constraints/roles.

Business process abstraction mechanisms should consider different aspects (Smirnov et al., 2012): the reason for abstraction that identifies the focus of abstraction, the conditions that should be satisfied and trigger the abstraction and the operations used for abstracting a process model to a more coarse-grained model.

Taking into consideration the above aspects the proposed abstraction mechanism focuses on providing process quick views based on defined rules that describe the conditions that should be satisfied for triggering the abstraction process and the abstraction operation (aggregation or elimination).

3 ABSTRACTION CONCEPTUAL DESIGN

In this section, we present the proposed conceptual design to process abstraction. The process abstraction mechanism is based on the use of transformation rules that are applied to business processes in a semi-automated way. The processes of the mechanism are defined with the use of the BPMN standard.

The overall process abstraction conceptual design is depicted in Figure 1 and is described in the following.

We consider that a user creates new business process models or modifies existing ones with the aid of a BPMN editor and stores them to a process repository. The process repository communicates

with the abstraction mechanism by sending existing process models and receiving the abstracted versions of the given process models.

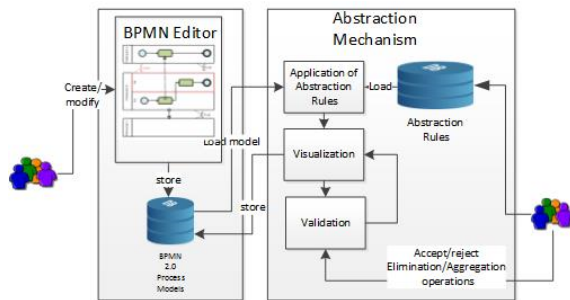


Figure 1: Conceptual design of Process Abstraction Mechanism.

The abstraction mechanism allows to apply abstraction rules to given models, visualize process elements candidate for abstraction, visualize statistics of process elements abstraction, validate abstraction and save abstraction results.

More precisely once a candidate process model of abstraction is defined in the abstraction mechanism, the mechanism communicates with the process repository and the process model is loaded into it. Then, the abstraction rules that are stored to a repository are used to trigger process abstraction. These abstraction rules presented in (Tsagakani and Tsalgatidou, 2015), are specially designed for process models defined using BPMN 2.0 and treat activities, paths, data objects, messages, lanes and artifacts as abstraction objects. These rules, when triggered, use aggregation and elimination as basic abstraction operations. Moreover, the rules are applied to the process model at hand while retaining the hierarchy in Figure 2.

The process abstraction is performed in a semi-automated way in the sense that before the finalisation of the process abstraction based on the applied rules, there is a validation phase where the user is involved. All the changes that are intended to be performed to the given process model are visualized both graphically and statistically. More precisely, each process element candidate for abstraction is clearly marked; then, it is upon the user to check its validity and accept or reject the abstraction. The resulted abstracted process model is also visualized along with some statistical results (e.g. the total reduction of activities, data, messages lanes etc.). Then, the abstracted process model is saved to the process repository where it is associated with the original business process model for future use.

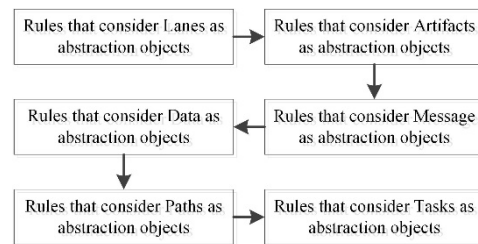


Figure 2: Abstraction Rule Hierarchy.

4 VALIDATION

In order to evaluate the proposed mechanism a detailed description of the ‘Loan Approval with no Collateral’ business process coming from the financial services industry was used. However in order to ensure confidentiality some details of the business process are omitted, whereas sufficient detail is provided in order to understand and illustrate the applicability of the presented conceptual framework.

Firstly, process models related to the ‘Loan Approval with no Collateral’ process were created based on the given detailed description, using the BPMN2Modeler that is an Eclipse-based graphical BPMN 2.0 model Editor (BPMN2Modeler, 2013). Therefore five process models were created (one model for each sub-process) – ‘Requirements Capturing and Loan Quote with no Collateral via Call Center’, ‘Loan Request Analysis and Pre-approval’, ‘Final Loan Approval’, ‘Processing of Loan Contractual Documents’ and ‘Loan Disbursement’. Then the rules that are defined for the specific abstraction mechanism were manually applied to the process models in the order that is presented in Figure 2 and validated. Finally the abstracted models were produced along with some statistical data that is explained below in this section.

Due to lack of space, in this paper we use for illustration purposes a small fragment of the ‘Loan Request Analysis and Pre-approval’ sub-process that is shown in Figure 3. It mainly presents the tasks needed to perform loan pre-approval, the departments and the roles involved, the messages that are exchanged and the data objects and artifacts that are associated to each task. The abstracted model that is the outcome after applying the defined abstraction rules of the proposed mechanism to the model (Figure3), is presented in Figure 4.

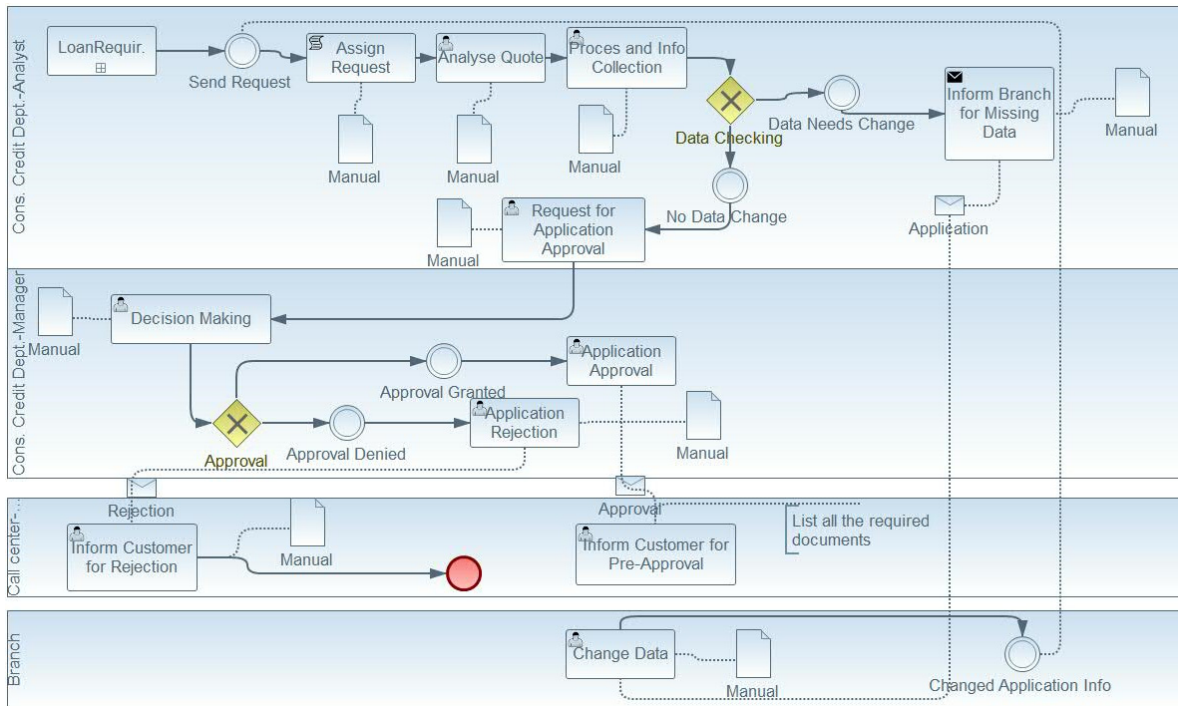


Figure 3: Fragment of the 'Loan Request Analysis and Pre-approval' process model.

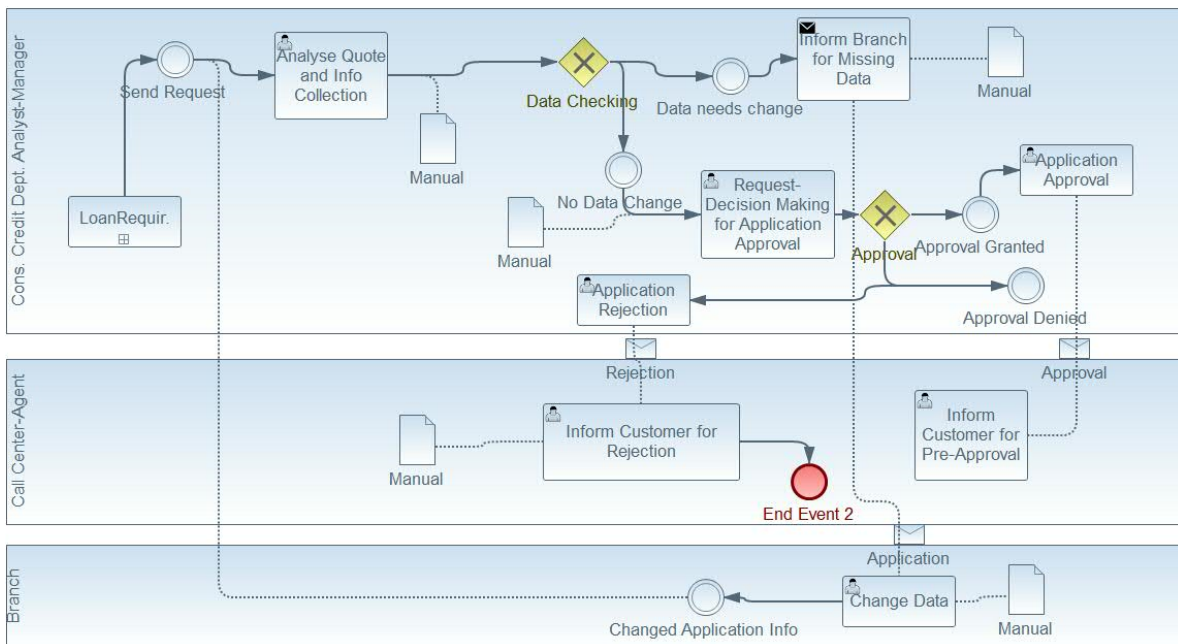


Figure 4: Abstracted Fragment of the 'Loan Request Analysis and Pre-approval' process model.

It can be observed that the two roles that co-exist in the same Consumer Credit Department are aggregated in the abstracted model. Also the script task is eliminated, the user tasks that are enclosed between the 'Send Request' event and the gateway

that follows are aggregated. Moreover the user tasks that are enclosed between the 'No Data Change' event and the gateway that follows are aggregated. The same applies to their data objects that are associated with the specific user tasks. The send task

‘Inform Branch for Missing data’ and the user tasks ‘Application rejection’, ‘Application Approval’, ‘Inform Customer for Pre-Approval’, ‘Inform Customer for Rejection’ and ‘Change Data’ are not eliminated in the abstracted model as they are associated with message exchange and are part of the process’s choreography. Finally the artifact associated with the user task ‘Inform Customer for Pre-Approval’ is eliminated.

The findings from applying the proposed abstraction framework to the ‘Loan Approval with no Collateral’ process are presented in Figure 5 where the total number of each process element that is affected by the abstraction process is presented before and after the abstraction. The findings illustrate that the activities that constituted the process were radically reduced from 70 to 43 that is 38.57% reduction. Moreover data objects show a 33.33% reduction, lanes 15.38% reduction, messages 22.22% reduction and artifacts 100% reduction.

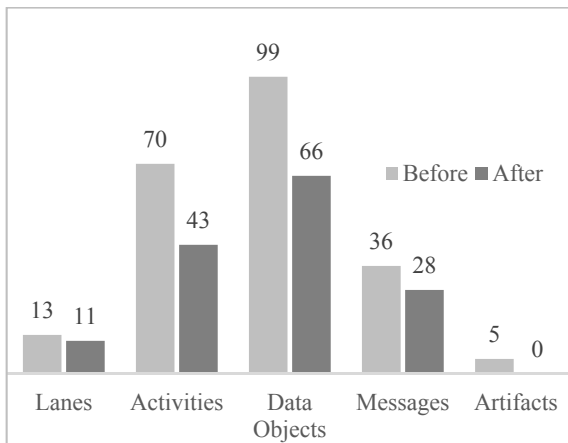


Figure 5: Total number of process elements before and after abstraction.

The business case used in this section to illustrate the value of our proposed mechanism and the findings produced seem to fulfil our research purposes that is to provide quick views of business processes while retaining the overall structure of the process and getting rid of insignificant details. Also prove that by introducing rules to the abstraction process related to process lanes, messages, artifacts and data objects (besides activities) may, to a great extent, reduce the size and complexity of the abstracted process models.

It is worthwhile mentioning that the proposed rule-based process abstraction has been also applied to other cases as well, with similar results. Some of the results have been briefly presented in (Tsagkani and Tsalgaidou, 2015).

5 RELATED WORK

There are a number research works that analyse business process models and base their abstraction mechanism based on transformation rules (Bobrik et al., 2007), (Cardoso et al., 2004), (Eshuis, and Grefen, 2008), (Günther and Van Der Aalst, 2007), (Meyer and Weske, 2012), (Pankratius and Stucky, 2005), (Polyvyanyy et al., 2008), (Sadiq and Orłowska, 2000) and (van Dongen et al., 2007). These research works not only analyse different process modelling notations and approaches but, in most of the cases, they also concentrate on activities as abstraction objects. An exception is the work by (Meyer and Weske, 2012) that considers data as abstraction objects. To the best of our knowledge the work of (Smirnov, 2009) is the only one that analyses BPMN 1.2, but it differs from our work as it suggests rules focused primarily on activities, while it only describes how other elements are influenced by activity abstraction.

The mechanism proposed in this paper targets business processes defined in BPMN 2.0 and is superior to other research works as it respects process’s overall structure and proposes abstraction rules that are focused on not only activities as abstraction objects but data, messages, artifacts, paths and process participants as well.

6 CONCLUSION AND FUTURE WORK

This paper presented the conceptual framework of a rule-based process abstraction mechanism. This framework preserves the structure of the initial model, focuses on process choreography and concentrates on certain abstraction objects (participants, activities, data objects, messages and artifacts). Furthermore the proposed framework was exemplified and validated using a business case coming from the financial services industry and the findings were discussed.

We are currently working on the detailed design and implementation of a tool based on the proposed conceptual framework of the abstraction mechanism. Besides, we are planning to further validate the proposed mechanism using more business cases.

REFERENCES

Bobrik, R., Reichert, M.U. and Bauer, T., 2007. Paramete-

- rizable views for process visualization.
- BPMN2 Modeler Project, 2013. Eclipse.org, software available at <http://projects.eclipse.org/projects/soa.bpmn2-modeler>.
- Cardoso, J., Sheth, A., Miller, J., Arnold, J. and Kochut, K., 2004. Quality of service for workflows and web service processes. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(3), pp.281-308.
- Chinosi, M. and Trombetta, A., 2012. BPMN: An introduction to the standard. *Computer Standards & Interfaces*, 34(1), pp.124-134.
- Eshuis, R. and Grefen, P., 2008. Constructing customized process views. *Data & Knowledge Engineering*, 64(2), pp.419-438.
- Günther, C.W. and Van Der Aalst, W.M., 2007. Fuzzy mining—adaptive process simplification based on multi-perspective metrics. In *Business Process Management* (pp. 328-343). Springer Berlin Heidelberg.
- Meyer, A. and Weske, M., 2012. Data support in process model abstraction. In *Conceptual Modeling* (pp. 292-306). Springer Berlin Heidelberg.
- OMG, B.P.M., 2011. Notation (BPMN) Version 2.0 (2011). Available on: <http://www.omg.org/spec/BPMN/2.0>.
- Pankratius, V. and Stucky, W., 2005, January. A formal foundation for workflow composition, workflow view definition, and workflow normalization based on petri nets. In *Proceedings of the 2nd Asia-Pacific conference on Conceptual modelling-Volume 43* (pp. 79-88). Australian Computer Society, Inc.
- Polyvyanyy, A., Smirnov, S. and Weske, M., 2008, November. Reducing Complexity of Large EPCs. In *MobIS* (pp. 195-207).
- Sadiq, W. and Orłowska, M.E., 2000. Analyzing process models using graph reduction techniques. *Information systems*, 25(2), pp.117-134.
- Smirnov, S., Reijers, H.A., Weske, M. and Nugteren, T., 2012. Business process model abstraction: a definition, catalog, and survey. *Distributed and Parallel Databases*, 30(1), pp.63-99.
- Smirnov, S., Weidlich, M. and Mendling, J., 2010. Business process model abstraction based on behavioral profiles. In *Service-Oriented Computing* (pp. 1-16). Springer Berlin Heidelberg.
- Smirnov, S., 2009, July. Structural aspects of business process diagram abstraction. In *Commerce and Enterprise Computing, 2009. CEC'09. IEEE Conference on* (pp. 375-382). IEEE.
- Tsagkani, C. and Tsalgatidou, A., 2015, October. Abstracting BPMN models. In *Proceedings of the 19th Panhellenic Conference on Informatics* (pp. 243-244). ACM.
- van Dongen, B.F., Jansen-Vullers, M.H., Verbeek, H.M.W. and van der Aalst, W.M., 2007. Verification of the SAP reference models using EPC reduction, state-space analysis, and invariants. *Computers in Industry*, 58(6), pp.578-601.

Cross-System Process Mining using RFID Technology

Christian Glaschke, Norbert Gronau and Benedict Bender

*Chair of Business Information Systems, especially Systems and Processes, University of Potsdam,
August-Bebel-Straße 89, Potsdam, Germany
{cglaschke, ngronau, bbender}@lswi.de*

Keywords: Process Mining, RFID, Production, Cross-System.

Abstract: In times of digitalization, the collection and modeling of business processes is still a challenge for companies. The demand for trustworthy process models that reflect the actual execution steps therefore increases. The respective kinds of processes significantly determine both, business process analysis and the conception of future target processes and they are the starting point for any kind of change initiatives. Existing approaches to model as-is processes, like process mining, are exclusively focused on reconstruction. Therefore, transactional protocols and limited data from a single application system are used. Heterogeneous application landscapes and business processes that are executed across multiple application systems, on the contrary, are one of the main challenges in process mining research. Using RFID technology is hence one approach to close the existing gap between different application systems. This paper focuses on methods for data collection from real world objects via RFID technology and possible combinations with application data (process mining) in order to realize a cross system mining approach.

1 INTRODUCTION

Since more than 20 years, business process management is the leading paradigm for organizing and restructuring of corporations and public entities. Although nearly all kinds of companies use business process management in certain areas, there are still some challenges that need further analysis. To name only a few, we can draw the reader's attention to the following issues (Gronau, 2016):

- Improving learning while performing a business process
- Improved usage of person-bound knowledge that is generated in or used during the business process.
- Establishment of PDCA cycles (plan-do-check-act) in process management to be able to detect deviations and to correct them without interrupting the business process.
- Typical business processes nowadays are supported by enterprise systems like ERP, CRM or SCM systems. Nevertheless, there are normally deviations between the intended process covered by the ERP and the process actually performed by the company (Gronau, 2015)
- Irrespective of the high degree of automation, in

most business processes, the human-machine interface is more important than ever because automated business processes inevitably have to be interrupted to allow the human being to decide or do something in order to propel the entire process. Today, however, the description of human interfaces is not intuitive at all.

- Last but not least, an increased real world awareness for nowadays available business process objects (like information, persons or customers' material) is needed. Also approaches to integrate these kinds of information into the respective processes are more necessary than ever.

Valid models of business processes are the basis for every optimization approach. Process mining has already shown some merits in some realms and is able to measure the actual performance of a business process - partially with the help of existing process models. Nevertheless, existing process mining approaches are limited in their scope to a single information system. Overcoming this barrier would not only bring the monitoring closer to the actual business process, it could also mean a big step ahead in the direction of process model discovery.

A main obstacle on the road to model improvement is the recognition of two entities with

different IDs in different information systems being one and the same (role, object, instance) in the real business process. For this purpose, i.e. to close this gap, this paper suggests to use RFID (Radio-frequency identification) technology.

2 PROCESS MINING WITH RFID

The focus of the present paper lies on the recognition of processes in heterogeneous application landscapes. In the course of digitalization, a lot of data on a company's business processes have become available. For instance, position vectors of real objects by means of RFID technology (Zhang et al., 2012), cyber-physical systems are able to realize process components autonomously and application systems have to be integrated over several levels in order to meet the requirements. To develop a system in this environment that allows for the reproduction and understanding of the business process is therefore the problem to be solved by the present work. This problem has already been defined and treated in other studies (Van der Aalst et al., 2012, Glaschke, 2015) and it aims at overcoming the system barriers. To achieve this aim, especially process mining is used, as this approach allows for an identification of process patterns through the analysis of a system's log files. The key idea of the present article is therefore to combine process mining with the data from RFID systems to thereby gain a comprehensive view on business processes.

In order to solve this problem, this paper will follow the Design Science Research Methodology (DSRM). According to Peffers et al. (2007) the DSRM involves six steps. Section one and two relate to the first two steps of the DSRM approach, namely the problem identification and definition of objectives. Section three discusses related techniques and technologies that could assist the cross-system based process mining. Section four corresponds to the design and development phase since it specifies the solution for cross-system based process mining. Furthermore, the second subsection demonstrates the solution with a theoretical example which relates to the demonstration phase. The evaluation of the approach needs to be conducted in an experimental setting for which the approach is described in the fifth section of this paper. With regard to the communication step the approach is presented in this paper, while the evaluation results will be published in further papers as well as necessary modifications of the approach.

Finally, the results will be critically discussed and

an outlook will be given. The DSRM (Design Science research methodology) seems appropriate since the approach is motivated by theoretical as well as practical issues and involves the design of an appropriate solution for the given problem and its evaluation.

3 RELATED WORK

For this paper, two research topics are of utmost importance. First, we will focus on process mining to identify the existing cross application approaches of this method.

The second relevant research topic is the technology of RFID. With respect to this issue, we will particularly focus on works that link these two approaches. Concerning the technology of RFID, we will especially explore the question of how the data is recorded in the application landscape and how it is provided for further processing.

At the end of this section, a brief summary of the existing research gap and the most important corresponding findings will be given.

3.1 Process Mining

A well-known approach for process discovery is the concept of process mining, which was developed by Van der Aalst and his research group at the Technical University of Eindhoven (The Netherlands). This approach uses log files from application systems (for instance ERP systems) to reconstruct process models. To be successful in that effort, the application system has to provide the information needed in a specific manner (i.e. as shown in Table 1).

Table 1: Example of a log file.

PID	Activity	Worker	Timestamp
452	registration	55	2011-12-24, 11:10:21
452	investigation	56	2011-12-24, 11:15:21
452	consulting	33	2011-12-24, 12:17:10
452	dismissal	55	2011-12-24, 12:47:11
453	registration	55	2011-12-24, 11:16:35
453	investigation	56	2011-12-24, 11:27:12
453	consulting	12	2011-12-24, 11:52:37
453	dismissal	55	2011-12-24, 11:59:54
454	registration	55	2011-12-24, 11:11:21
454	investigation	55	2011-12-24, 11:15:21

An important component needed for this listing of

process instances is the process identification number (PID). This number is used to create a process diagram based on more than just one process instance (Van der Aalst, 2012). In the background, petri networks are used here (i) to allow for the generation of process diagrams, (ii) to describe the different conditions of the process and (iii) to create a graph for visualization purposes and analysis (Van der Aalst, 2011a; Accorsi, 2012).

In an article from Thiede and Fuerstenau (2016), an extensive literature review on this topic is presented. Therefore, the top 20 AIS Journals between 2004 and 2015 are analysed. Additionally, literature from production and organisational journals have been screened for process mining and RFID implementation publications. It could be concluded that the usage of RFID to close the gap has not been researched so far.

As can be deduced from Thiede and Fuerstenau (2016), many research projects concentrate on the field of “digital services” and among these especially on the subtype of single systems. Thiede and Fuerstenau (2016) define the concept of “service” broadly as “the application of specialized knowledge skills through deeds, processes, and performances for the benefit of customers (Vargo and Lusch, 2004, p. 2). For instance, a customer buying an article in a web shop (digital service) receives it by a dispatching service (non-digital).”

Out of this studies the work on Cross-System Process Mining is analyzed. There are different approaches with wide scattered areas of application. Now we describe the most important ones. The first approach deals with data from smart environments. The main result is an algorithm to mine frequent patterns (Wen, 2015).

Another work deals with the theme of end to end Process Management. The authors describe implications for theory and practice based on the end to end process management approach. Especially the technological challenge is highlighted in the investigation of end to end processes (Madder et al., 2014).

Cross-organizational observations of processes, however, have so far only been presented by very few studies and also these work primarily focused the aspect of workflow synchronization. An example of this specific application of process mining is given by Zeng et al., (2013).

Zeng et al., (2013) define in their work several approaches for the synchronization of different log files:

1. Coordinated with synchronized activities.
2. Coordinated with messages exchanged.

3. Coordinated with shared resources.
4. Coordinated with abstract procedures.

These four coordination patterns can generally be used to link the different log files from various companies. Nevertheless, this solution proposal is based on the assumption that the respective processes remain closed systems only linked to each other at some specific points. As examples hereof, we can cite the cases of goods delivery (activity) or document exchange (messages). Based on these patterns, rules can be defined that determine when exactly a synchronization takes place in two log files or models.

3.2 RFID

Today and due to digitalization, data on numerous business objects is available. So, for instance, the position vectors of an object can nowadays be calculated in real-time due to RFID technologies (Yingfeng et al., 2012). These data can then be used to analyze the goods flow like demonstrated, for example, in Jakkhupan (2012). For this purpose, the objects involved in the process are first tracked and then the respective data is saved.

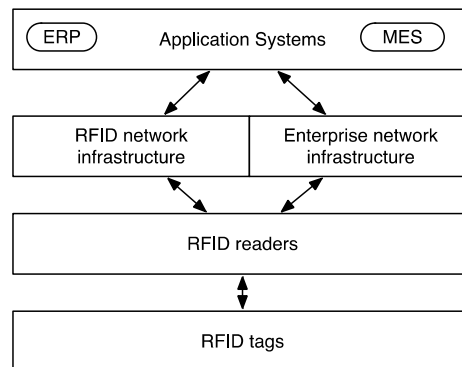


Figure 1: RFID infrastructure (Krishna, 2007).

In a typical RFID infrastructure as shown in Figure 1, the readers send unprocessed data to an application system. In other architectures, middleware has to be used for this purpose (Abad et al., 2012). Furthermore, cross system communication is made possible by writing collected information on the RFID tag to be read out and further processed at other points in the process (Finkenzeller, 2015).

A combination of both technologies, i.e. of RFID and process mining, has already been used by Zhou (2010) for the reconstruction of health care processes. For this approach, the techniques of process mining were applied to both, log files and the RFID

infrastructure. In doing so, it could be evidenced that the log files of RFID infrastructures are also suitable for being used in the context of process mining applications and techniques (Gonzalez, 2006).

Fernandez et al., (2015) use a similar approach working with indoor location systems to collect data on different business objects in order to draw conclusions concerning the corresponding processes. For this second step, the authors also work with process mining techniques.

3.3 Evaluation

Particularly from a reading of the work of Zeng et al., (2013), the question arises of how the present paper will distinguish from previous research. Synchronization of workflow models has already been described extensively in literature. Hence, this article primarily differs, on the one hand, with respect to the investigated area, as here company internal processes are investigated. On the other hand, in our approach, the different workflow models from the single application systems shall be integrated and not just synchronized. Thus, we aim at establishing an end-to-end business process that is interconnected across the different system boundaries. To achieve this goal, it is for example suitable to work with the shared resources approach (Chapter 3.2), with the respective data being tracked – in our case – by RFID systems. In choosing this solution, it is not necessary that both systems use the same denomination or ID for one and the same object (Chapter 4).

4 CROSS SYSTEM PROCESS MINING

In order to achieve the target to extract reliable and encompassing models of process execution, it is nowadays necessary to link information from multiple information systems. This is due to the fact that business processes involve different functional areas within a company which are usually using highly specialized software applications in order to fulfill their tasks. As a result hereof, companies' application landscapes involve many different application systems, each executing a little part of the overarching process (Fuerstenau and Glaschke, 2015).

4.1 Requirements and Objectives of the Procedure

Established techniques like process mining are con-

ceptualized to evaluate information from single information systems. Typically, for this purpose, log files from one information system are analyzed. In this procedure, firstly log entries are grouped according to their affiliation (e.g. by assorting all log entries that belong to a certain PID), secondly dependencies and chronological consecutions are analyzed and thirdly and finally all the aforementioned bits of information are aggregated in a workflow model.

In order to face now the challenge to model complex cross-system processes, the concept of process mining shall be complemented by a linking technology in order to be able to model complete business processes that are not limited to a single information system. This linking technique shall allow for a combination of the different sub-process models that could be extracted from the different information systems participating in the overarching process.

One essential precondition of this approach is that each information system that forms a part of the overall process is connected with the RFID infrastructure. This is necessary for the PID from the information system to be passed to the RFID infrastructure.

Another necessary prerequisite for this solution is that the PIDs from the different information systems are unique, i.e. that they are working with different number ranges. Only in this way, each process instance can be identified unequivocally.

Besides, also the given RFID tag must identify the objects unequivocally by ignoring the sharing or merging of RFID tag numbers.

Thus, our aim is to facilitate the integration of various log files and the therein contained process instances as illustrated in Figure 2.

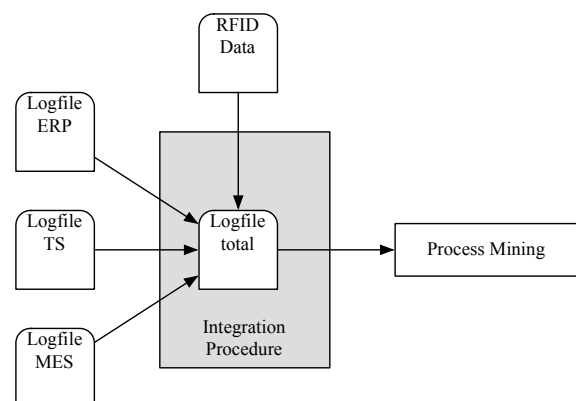


Figure 2: Fundamental procedure for log file integration.

The combination of information system log files

with RFID data should reveal the overarching process model which is composed of single system specific process models that are combined in the appropriate order. One of the major challenges in this context is to identify corresponding sub-processes in the different information systems. In order to face this challenge, a distinct linkage between the different sub-process models is needed.

Just like in case of process mining, configurations have to be made within the information systems for the created log files to provide an uninterrupted description of the process instances. To achieve this, each generated instance has to pass the current process ID to the RFID infrastructure at known interconnection points. This PID is then saved on the respective RFID tag. To make this possible, scanning processes must be performed at specific process steps. Ideally, these scans will take place at the beginning and end of a process.

In order to realize the linkage between the different information systems, two kinds of physical components are needed. The first component are containers that physically accompany the different sub-process steps through the entire process. This kind of containers is set up in the first process, which is executed in the ERP system. At this point, an empty container is equipped with an RFID-tag that has a unique ID.

The second component is the aforementioned infrastructure of RFID gates, which is used to scan the containers accompanying the different process steps.

Figure 3 illustrates how the individual scanning processes are registered as entries on the RFID tag. To allow for this kind of documentation, the involved application systems have to be customized accordingly. This customization procedure is already standard in case of process mining applications (Van der Aalst, 2011b). This technique, however, additionally requires that when scanning the RFID tags, the data is passed and stored. This also requires

the appropriate configuration of the participating application systems.

Yet, this solution proposal is just one variant. Another method would consist in passing the unequivocal RFID tag ID to the information systems to be saved there. In this case, the tag ID would have to be written to the logs of the application systems. Another possible approach would be to log the identity within the RFID architecture. As here the respective functionality needed to log the scan processes is given, this variant will be favored in the context of our validation.

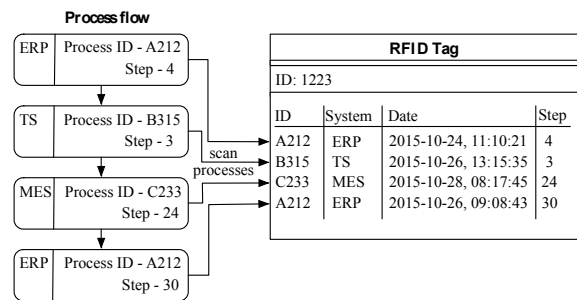


Figure 3: Data structure on the RFID tag after scanning.

4.2 Approach for an Integration Procedure

Although the here presented solution proposals seem to be convincing, there is a significant gap between the processing through information technology and reality. Whereas in an information system tasks can already be performed without the necessity of a concrete processing object with RFID tag to exist, in reality and as shown in figure 4, there will always be overlapping time entries in the log files.

As strategy for our integration procedure, we therefore have to take the following steps: The process steps will have to be passed to the result log following their chronological order and in doing so,

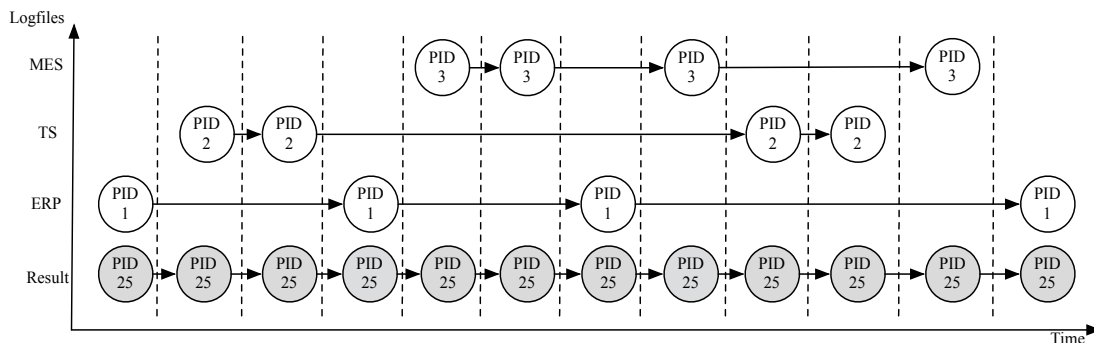


Figure 4: Basic procedure for log file integration (for one process instance).

the PIDs from the application systems will have to be replaced by the PID on the RFID tag. For this purpose, the algorithm goes through the entries on the RFID tag, systematically replacing the PIDs in the log files, to afterwards copy the lines into one new contiguous result log file.

The process mining procedure which further processes the result log file then uses the corresponding algorithms in order to reconstruct the workflow with the aid of the time entries. Associated examples and algorithms for this procedure are presented by de Medeiros et al. (2003). For the validation of this approach, an appropriate data model is needed which is illustrated in Figure 5. Each RFID reader is assigned to one place and one application system. The RFID readers generate log entries, which, in turn, are assigned to just one business object. In our case, it is the read data and the written data that is logged. Then, the software passes the reading and writing processes to the assigned application system.

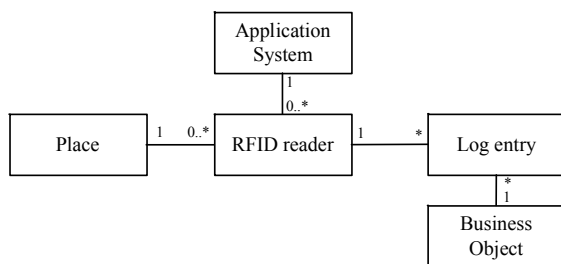


Figure 5: Data Model of RFID Management Software.

Furthermore, we developed a process to be used for validation that can also serve as reference for the obtained result models. This process was designed to reflect the typical production process of a manufacturing company and it comprises the following phases:

To begin with, one order with several items results in different manufacturing jobs in the ERP system. Next, the corresponding components are commissioned and the production container is packed. Afterwards, the order is scanned for the first time and is passed on to the transport system (TS), which moves the container to the first free workstation. Subsequently, several production steps are carried out in MES (Manufacturing Executin System). At different process steps, the container is scanned for identification purposes. In order to measure the product properties, the data is handed over to a quality management system and the measurement results are documented. After this step, the container is transported to the warehouse or distribution center.

Even though the approach seems to solve a range of problems with regard to the cross-system process mining, it has to be remarked that this type of aggregation cannot be used for all types of processes. The applicability is limited to processes where the physical parts, that are subsumed in the container, are scanned at all of the involved information systems or the data is passed, at least once, to each of the participating information systems of the process. This is due to the fact that the combination technique relies on a bridging mechanism based on part handover.

5 VALIDATION

In order to validate the concept for cross system process modeling, several steps are necessary, which need to be executed subsequently.

As a first step, an appropriate model composition procedure needs to be realized in order to integrate the different log files into one overarching log file. At this point, and similar to the proceeding in process model generation, the distinction of relevant and irrelevant information determines the quality of the composed overarching process model. This composition process is supported by the information from the RFID management software and, in particular, by the unique IDs that are allocated by the software component.

As a second step with regard to validation, the experiment itself needs to be set up. Therefore, a production setting is needed which allows for the representation of the formerly defined process. This stage involves physical components, such as some kind of production machines, as well as the related software infrastructure, which is composed of the different systems mentioned in the process. Finally, the RFID management software as well as the RFID infrastructure including the containers have to be set up.

For the validation of our approach, a university Industry 4.0 Laboratory is used. This laboratory is equipped with several application systems like, among others, an ERP and a MES system from the company Asseco Solutions AG. Additionally, different infrastructure components like robots, transportation systems and a RFID infrastructure are available. The manufacturing process can thus be simulated through configuration of the components or it can be actually performed (figure 6).

The RFID infrastructure consists of stationary and portable RFID readers with which RFID tag information can be read and written. For the simple configuration of the RFID infrastructure, we have a

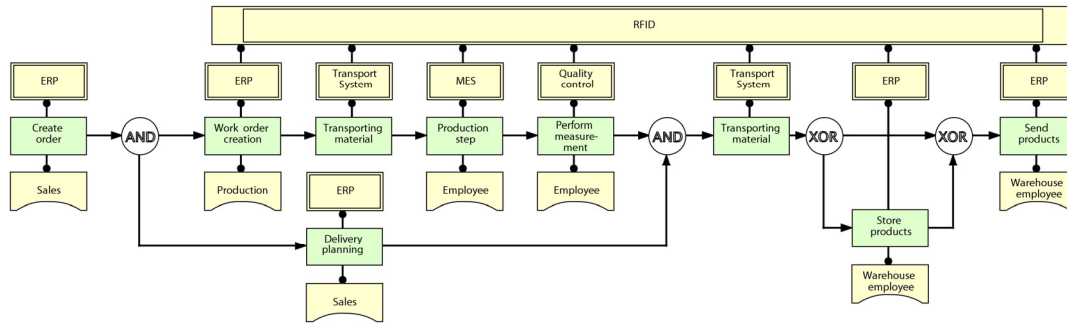


Figure 7: Example process with RFID support.

management software that is used as middleware in the laboratory plant.



Figure 6: Physical validation environment.

Besides, delivery planning takes place while the production is in progress. Based on the thereby elaborated scheduling list, the TS decides where to transport the products (warehouse vs. distribution center). These steps are also logged and get assigned the PID from the ERP (Enterprise Resource Planning) system.

For validation the process should be executed several times with different variations. Especially the different mining algorithms from the process mining software (Claes and Poels, 2012) will here have to be tested and the respective results will have to be compared. The validation will be successful when the elaborated models actually reflect the process shown in Figure 7.

6 CONCLUSIONS AND OUTLOOK

In this paper, it could be demonstrated that a simple and accessible method can close the gap between the different systems and processes. For this purpose, the

concept of shared resources (Zeng et al., 2013) was applied, which is already used in cross-organizational approaches. Furthermore, it could be shown that the RFID technology does already provide a solution concept. Thus, we have succeeded in presenting a cross system integration procedure.

If the results are achieved within the laboratory environment, the question arises of whether the approach will also function in practice and in more complex environments with massive parallel operations. All in all, the question remains of what factors are responsible for the quality of results. Here, on the one hand, the quality of the log files from the application systems will definitely have to be taken into consideration and, on the other hand, the number of iterations needed to identify parallel and independent process steps will be of importance. The validation results will be published in further works.

In this paper, a simple setting was chosen to demonstrate the benefits of the technology combination of process mining and RFID. Future work must continue to follow this trend. It is possible to track more than one and different business objects. The application system provides further information, which can be used for the enrichment of process model analysis. An example - in this case products were anonymous, but they can be categorized in future according to the product category. So it is possible to structure the mined process model.

To follow the idea of end to end processes we have to consider the supplier processes and other support processes like maintenance. Therefore laboratory environments with real applications systems and physical components can help, to establish the techniques in practice by showing practical solutions and related benefits.

REFERENCES

Abad, I., Cerrada, C., Cerrada, J. A., Heradio, R., Valero,

- E., 2012. *Managing RFID sensors networks with a general purpose RFID middleware*. *Sensors*, 12(6), 7719–7737.
- Accorsi R., Stocker T., 2012. *On the Exploitation of Process Mining for Security Audits: The Conformance Checking Case*. *ACM Symposium on Applied Computing*. doi:10.1145/2245276.2232051.
- Claes, J., Poels, G., 2012. *Process mining and the ProM framework: An exploratory survey*. *Business Process Management Workshops*. Springer Berlin Heidelberg.
- De Medeiros, A. A. K., Van Dongen, B F, Van der Aalst, WMP., Weijters, A.J.M.M., 2003. *Process mining: extending the α -algorithm to mine short loops*. *On the move to meaningful internet systems*, Springer Berlin Heidelberg, 389-406.
- Fernandez-Latas, C., Lizondo, A., Monton, E., Benedi, J. M., Traver, V., 2015. *Process Mining Methodology for Health Process Tracking Using Real-Time Indoor Location Systems*. *Sensors*, 15(12), 29821-29840.
- Finkensteller, K., 2015. *RFID-Handbuch: RFID Handbook: Fundamentals and practical applications of transponders, contactless smart cards and NFC*. (in german) Carl Hanser Verlag, Volume 7.
- Fuerstenau, D., Glaschke, C., 2015. *Weighting of Integration Qualities in IS Architectures: A Production Case*. Paper presented at the ECIS 2015, Germany.
- Glaschke, C., Gronau, N., 2015. *Automated Process Model Discovery - Limitations and Challenges*. Paper presented at the International Symposium on Business Modeling and Software Design.
- Gonzalez, H., Han, J., Li, X., 2006. *Mining Compressed Commodity Workflows from Massive RFID Data Sets*. 15th ACM International Conference on Information and Knowledge Management, ACM, 162-171.
- Gronau, N., 2016. *Business Process Management in industry and administration*. (in german), Gito Verlag, Berlin, 2016.
- Gronau, N., 2015. *Trends and Future Research in Enterprise Systems*. *Lecture Notes in Business Information Processing*, Volume 198, 271-280.
- Jakkhupan, W., Archint, S., Li, Y., 2012. *Business process analysis and simulation for the RFID and EPCglobal Network enabled supply chain: A proof-of-concept approach*. *Network and Computer Applications*, 34(3), 949-957.
- Krishna, P., Husalc, D., 2007. *RFID Infrastructure*. *IEEE Communications Magazine*, 45(9), 4-10
- Maddern, H., Smart, P. A., Maull, R. S., Childe, S., 2014. *End-to-End Process Management: Implications for Theory and Practice*. *Production Planning & Control*, 25(16), 1303-21.
- Peffer, K., Tuunanen, T., Rothenberger, M., Chatterjee, S., 2007. *A Design Science Research Methodology for Information Systems Research*. *Journal of Management Information Systems*. 24(3), 45-77.
- Thiede, M., Fuerstenau, D., 2016. *The Technological Maturity of Process Mining: An Exploration of the Status Quo in Top IS Journals*. Paper presented at the MKWI 2016, Germany.
- Van der Aalst, WMP., Accorsi R., Ullrich M., 2012. *Process Mining*. Retrieved. (in german) April 29, 2015, from <http://www.gi.de/nc/service/informatiklexikon/detailansicht/article/process-mining.htm>
- Van der Aalst, W., 2011a. *Process Mining - Discovery, Conformance and Enhancement of Business Processes*. Springer.
- Van der Aalst, W., 2011b. *Service Mining: Using Process Mining to Discover, Check, and Improve Service Behavior*. *IEEE Transaction on Service Computing*, 6(4), 525-535.
- Vargo, S., Lusch, R., 2004. *Evolving to a New Dominant Logic for Marketing*. *Journal of Marketing*, 68(1), 1-17.
- Wen, J., Mingyang Z., Zhiying W., 2015. *Activity Recognition with Weighted Frequent Patterns Mining in Smart Environments*. *Expert Systems with Applications*, 42, 6423-32.
- Yingfeng, Z., Jiang, P., Huang, G. Qu, T., Zhou, G., Hong, J., 2012. *RFID-Enabled Real-Time Manufacturing Information Tracking Infrastructure for Extended Enterprises*. *Journal of Intelligent Manufacturing*, 23, 2357-2366.
- Zeng, Q., Sun, S. X., Duan, H., Liu, C., Wang, H., 2013. *Cross-Organizational Collaborative Workflow Mining from a Multi-Source Log*. *Decision Support Systems* 54, 1280-301.
- Zhou, W., Piramuthu, S., 2010. *Framework, strategy and evaluation of health care processes with RFID*. *Decision Support Systems*, 50(1), 222-233.

Towards a Service-Oriented Architecture for eVoting

Boris Shishkov^{1,2} and Marijn Janssen³

¹*Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Sofia, Bulgaria*

²*IICREST, 53 Iv. Susanin Str., Sofia, Bulgaria*

³*TBM – TU Delft, Jaffalaan 5, Delft, The Netherlands*

b.b.shishkov@iicrest.org, M.F.W.H.A.Janssen@tudelft.nl

Keywords: eVoting, Conceptual Model, Requirements, Architecture.

Abstract: The latest advances in the Information and Communication Technology (ICT) are changing our society, but have different implications on different domains. Some domains, such as the digital content –based businesses, are enjoying (almost) full ICT utilization whereas other domains, assuming physical and/or societal and/or “intuitive” inputs, are much less successful in terms of digitization. Voting using digital technology (or “eVoting” for short) is in between those domains since: (i) the mere process of voting is a very good “candidate” for digitization but at the same time (ii) the “surrounding” societal aspects are often difficult to “frame” as Internet-based services. (i) can be seen from the “voting through computer” observed in several European countries while (ii) can be seen from the lack (to date) of technology-enabled systems completely supporting the voting process and its related aspects. Further, the conceptualization and implementation of any voting system is to originate from legislation – this makes the goal of resolving (i) + (ii) even more challenging. Hence, to benefit from ICT, the question remains what should be done and how it should be done. The step from legislation to requirements and implementations taking into account socio-technical aspects, is crucial for the successful realization of eVoting. Despite its relevance, this has been given hardly sufficient attention in literature. This void is addressed by the current position paper; the contribution of the paper is two-fold: we firstly propose a general technology-independent conceptual model on voting and on this basis, we propose requirements for (partially) digitizing this process. Requirements are dependent on the societal context and therefore we opted for focusing on one particular EU country where the transition to eVoting is currently under discussion. We have planned as future research to reflect the identified requirements into architectures and implementations, and to get experts’ feedback on this.

1 INTRODUCTION

Advances in the Information and Communication Technology (ICT) are changing our society, and each domain has its own “way” of utilizing computer / Internet –based services. Some domains, such as the digital content –based businesses, are enjoying (almost) full ICT utilization. Other domains, assuming physical and / or societal and / or “intuitive” inputs, are less successful in terms of digitization. Technology-enabled voting (referred to as “eVoting” in the current paper) is in between those domains since: (i) the mere process of voting is a very good “candidate” for digitization but at the same time (ii) the “surrounding” societal aspects are often difficult to “frame” as Internet-based services. (i) can be seen from the “voting through computer” observed in several European countries while (ii) can be seen from the lack (to date) of a technology-enabled

system completely supporting the voting process and its related aspects (Scammell, 2016). Hence, the eVoting domain should be conceptualized from a socio-technical perspective. Further, the conceptualization and implementation of any voting system is to originate from legislation – this makes the goal of resolving (i) + (ii) even more challenging. Thus, the question remains what should be done and how it should be done, in order to both benefit from ICT advances and stay adequate in terms of societal integrity. Meeting all requirements is crucial for eVoting, as failure in voting is not an option in a democratic society. Still, the step from legislation to requirements and implementations is essential for the successful realization of eVoting. Nevertheless, despite its relevance, this particular step has been given hardly sufficient attention in literature. This justifies our work, reported in the current position paper – to conceptualize the voting process and derive

requirements (and guidelines) for eVoting. Hence, the contribution of the paper is two-fold: we firstly propose a general technology-independent conceptual model on voting (by “voting” we mean the political voting for parliament, president, mayor, and so on, taking place in different countries, as opposed to for example corporate voting for board of directors or other kinds of voting) and on this basis, we propose requirements for (partially) digitizing this process. Still, since those issues are inevitably based on a particular societal context and because we need to be “concrete” in our modeling activities, we have decided to base our work on the situation in one particular EU country where the transition to eVoting is currently under discussion. We have planned as future research to reflect the identified requirements into architectures and implementations, and to get experts’ feedback on this.

In the remaining of this paper: In Section 2 we present theoretical background and in Section 3 we present our research focus and propose a conceptual model on voting accordingly. On this basis, we derive requirements (Section 4). Then, in Section 5, we discuss the next step, namely reflecting the identified requirements in architectures and implementations, and we propose some general guidelines in this regard. Finally, we present the conclusions in Section 6.

2 THEORETICAL BACKGROUND

We propose a technology-independent model on voting, based on the theories of *LAP – Enterprise Ontology* (Dietz, 2006), *Organizational Semiotics* (Liu, 2000), *Workflow Management* (Van der Aalst, 2011), *Service-Oriented Computing* (Papazoglou, 2012), and *Conceptual Modeling* (Insfran et al., 2002). Those are briefly outlined in the current section.

2.1 LAP – Enterprise Ontology

The *Language-Action Perspective* (“LAP”) theory (Shishkov et al., 2006) emphasizes the importance of interaction and communication, recognizing that language is not only used for exchanging information, as in reports (for example), but that language is used also to perform actions, as in promises or orders (for example). Such actions are claimed to represent the foundation of communities and organizations / enterprises. This relates to the *white-box model* of an organization that is of key importance for building

valid enterprise ontologies – this model acknowledges actors (the entities fulfilling corresponding actor-roles) who may be involved not only in *production acts* (for example: deliver a pizza) but also in *coordination acts* (for example: promise a delivery), and those acts may be of relevance to three perspectives of an organization, namely: *documental* (documents being created and used, for example), *informational* (customer enters PIN in order to realize a bank transaction, for example), and *essential* (the bank transaction itself, for example). Finally, *Enterprise Ontology* considers a generic interaction atomic pattern, claiming that any complex interaction can be decomposed in such pattern primitives and there are always two roles, namely customer (the one who initiates anything, for example – order something) and producer. There is a *request-promise-execute-state-accept* actions sequence between them and it can be reflected in a success layer and also a failure layer, as well as discussion layer, in between. For more information on Enterprise Ontology, interested readers are referred to (Dietz, 2006).

2.2 Organizational Semiotics

Organizational Semiotics (OS) addresses a number of concepts, such as *sign* and *affordance*, as essentially useful in modeling a (real-life) system and adequately considering relationships and meanings. Often what we observe goes beyond the primary “appearance” – for example, one could hold a Rolex pen not only as a means of writing but also as a way of demonstrating wealth (this is a sign). As for the affordance concept, it relates to potential abilities (for example: a book affords to be borrowed). Those concepts and also other OS concepts, allow for building complex models that reflect both *semantics* and *norms* (rules), and that is reflected in the widely popular OS norm pattern:

```
whenever <condition>
if <state>
then <agent>
is <deontic operator>
to <action>
```

The OS norm pattern is considered useful in modeling relationships among entities, in the context of a business process (Shishkov et al., 2006). For more information on OS, interested readers are referred to (Liu, 2000).

2.3 Workflow Management

It is claimed that any business process can be viewed

as a collection of processes, where a *process* can be described as “a set of identifiable, repeatable actions which are some way ordered and contribute to the fulfilment of an objective”; typical process patterns are *sequence, parallelism, split*, and so on.

Workflows play useful role in modeling business processes and those models can be enriched in terms of OS norms and / or entities information. For more information on Workflow Management, interested readers are referred to (Van der Aalst, 2011).

2.4 Service-Oriented Computing

Web services appear at high-level to be (dynamically) composed by users, hiding thus their underlying technical complexity – this complexity is with the software components who are implementing the corresponding service(s). *Composability, traceability, and interoperability* are hence of crucial importance in web service provisioning. More information on those issues can be found in (Papazoglou, 2012).

2.5 Conceptual Models

We consider a *conceptual model* as an abstraction with regard to the real world. As for information (IT) systems being developed, they inevitably need to be based on such abstractions because among other things, an information system is about the automation of real life processes. Nevertheless, the value of conceptual modeling efforts often remains unclear (Insfran et al., 2002) and often software engineers do not know whether a conceptual model represents the user’s requirements. Finally, we argue that in order to be useful in such a context, a conceptual model is to adequately capture the *real life functionalities* that would be (partially) automated, as part of the development of information systems. In order to address this and taking especially an *eVoting perspective*, we:

- a. would not keep the conceptual model too abstract (we would have it reflect a particular (voting) context);
- b. would separate the technology-independent conceptual modeling from the IT-inspired requirements identification and specification;
- c. would make particular assumptions in order to “take out” of consideration aspects with no relevance to what could actually depend on the IT system as such.

We argue that this would lead to establishing a more explicit role of conceptual modeling with regard to

the development of information systems (particularly in the eVoting context) and also to guaranteeing that conceptual modeling is aligned to requirements engineering.

For this reason, we will firstly outline (in Section 3) the “modeling context”, namely the situation (regarding eVoting) in one EU country (Sub-Section 3.1), secondly, we make our assumptions – in line with what was mentioned above (Sub-Section 3.2), thirdly, we propose a technology-independent model (Sub-Section 3.3), and we extend this, by identifying and specifying requirements (Section 4).

3 THE VOTING MODEL

As already mentioned, in this section, we consider the societal context, we present assumptions, and in the end – the technology-independent model on voting.

3.1 Societal Context

For the sake of “grounding” our research to a particular societal context / case, we went for considering a concrete eVoting context, namely the situation in *Bulgaria* (Konstantinov et al., 2009). The latest developments in Bulgaria with regard to eVoting are considered “representative” because they reflect the current EU visions on that issue, trying to balance between the scepticism (observed in Germany and other countries) and the “success stories” (observed in Estonia and other countries). Moreover, the current legislation changes and initiatives in Bulgaria reflect the latest ICT developments which was not the case in the countries who have introduced eVoting several years ago. For this reason, it is not surprising that eVoting dominated an October’15 referendum in Bulgaria (Plevneliev, 2016); then the majority of Bulgarians voted in support of “IT-enabled voting” and the “pro” campaign was backed by particular stated eVoting *public demands* that may (eventually) be reflected in corresponding legislation changes; some of those demands are presented and briefly discussed below:

- *secrecy of vote*, possibly achieved through anonymous credentials, such that not even the system “knows” how a person has voted;
- *cost adequacy*, possibly achieved through smart decisions rather than posh hardware that would generate future “dependencies”;
- guarantee against *violations* with regard to the way the system works;
- guarantee against *manipulations* of the final

- voting results;
- support of *secure communication* between the computers and the servers that is to be possibly cryptography-enriched.
- *controllability* - any third parties should be able to "verify" that the system is working properly;
- guarantee that each vote has been counted and that the person who had voted would *not be allowed to vote again*;
- *fault-reaction* is to be established as guarantee that even if the system (partially) crashes, it would recover and this would not affect its storage and processing functions;
- *ease of use* even by persons who are not of high computer literacy;
- no need for extra *qualification* of the election authorities.

Obviously, such a public demands list cannot be exhaustive and it is also inevitably unstructured. Moreover, those "demands" reflect a mixture of things – from FUNCTIONAL to NON-FUNCTIONAL, from CONCEPTUAL to TECHNICAL, and so on. For this reason, we should "extract" some information in support of our technology-independent functional conceptual model, and we should as well extract information in support of the requirements' identification to be addressed in Section 4. Hence, in the following list, we refine the demands, to achieve *input* for our conceptual model, abstracting from all non-functional and technical aspects:

- secrecy of vote;
- fair reflection of votes;
- fair communication of intermediary results;
- fair counting of votes;
- the voting - easy for all having rights to vote.

Thus, in line with those demands, any person with voting rights in Bulgaria should be able to vote. This assumes that no special qualification / skills are needed, in such a way that both his/her vote is fairly reflected in the voting system and the secrecy of his / her vote is guaranteed (this means that the vote should be counted but it should not be linked in any way to the person who has executed the voting). Finally, it should be guaranteed that all intermediary results are properly communicated to the voting "central" and in the end – all votes are correctly counted, to guarantee adequate political representativeness.

In Bulgaria, people vote for parliament, for presi-

dent, for local authorities, and also in referenda. Voting is preceded by a campaign. During the campaign, candidates / parties have the right to register for the vote and also to campaign for / promote their ideas. This is supposed to stimulate people to analyse the current situation and the (potential) impact of different political influences. Then, the campaign stops and the day before the elections is made free of campaigning – to avoid pressure on voting in a certain direction. This is followed by the voting "day" lasting 12 hours; during this 12-hour period, people have the right to go and vote (or not to do so), and once voting is done, it is done – it is impossible to return back to the voting station and claim changing mind and voting again. Further, when the voting "day" is over, votes are counted in each voting station and this would generate the so called "raw" results, reflected in a list of candidates and corresponding numbers. Finally, all such lists are "brought together" as the source for calculating the overall results per country and / or per town and / or per county, and so on.

3.2 Assumptions

The above is the starting point for developing our conceptual model. For this reason, we should know *what to keep "in"* and *what to leave "out"*, and this decision is inevitably technology-driven because we leave out things that cannot be (at all) dependent on what the technology can offer. For example, if Steve is going to vote from home via computer and his father is standing behind him, influencing his vote, then we cannot do anything about this no matter how advanced technology is – this is just matter of each person's observing and upholding the rights each of us has. We thus make several assumptions: (i) issues of societal relevance which are nonetheless beyond the information system control are outside the scope of this research (consider the above example of Steve's voting). (ii) We assume sufficient IT literacy among the population, realizing nevertheless that there are poorly developed regions where this is not yet the case. (iii) We assume a democratic country in which governments do not manipulate the election process. The above assumptions maybe (partially) IT-inspired but even so, the assumptions have straightforward impact with regard to the technology-independent conceptual model because they lead us to what to leave out of the model.

3.3 The Voting Model

In order to keep (as promised) the model abstract (thus simple) but also properly focused, we: (i) apply only two concepts, namely actor-role and relationship; (ii) take into account what was presented in Sub-Section 3.1 and Sub-Section 3.2.

Our Actor-Role (AR) concept is consistent with Enterprise Ontology (see Section 2), suggesting that ‘actor’ is the (human) entity executing a particular task while ‘actor-role’ is about specifying the task itself, abstracting from who exactly is fulfilling this. For example, if a Professor is sending fax, then (s)he is doing something that is part of what the Secretary normally does and for this reason, determining the ‘actor-role’ here points to the label ‘SECRETARY’ (not ‘PROFESSOR’). Our Relationship (R) concept is consistent with the SDBC approach (Shishkov, 2005) and is about whether or not collaboration is needed between two ARs that is necessary for one or both of them to deliver what they have to deliver.

Hence, we have identified the following ARs in the context of what was presented in this section:

AR1 – **CAMPAIGNER**: the one(s) campaigning in favour of a particular policy / party / vision and influencing the people in that way;

AR2 – **VOTER**: the one(s) voting for parliament / president / ... and thus executing basic rights in the country;

AR3 – **PRIMARY COUNTER**: the one(s) counting the votes in a particular voting station;

AR4 – **SECONDARY COUNTER**: the one(s) aggregating the final result, by putting together the voting results from the voting stations;

AR5 – **ORGANIZER**: the one(s) organizing the voting process and supporting all above-mentioned accordingly;

AR6 – **CONTROLLER**: the one(s) controlling all above-mentioned;

7 – **SYSTEM**: even though this is not an actor-role, we have to somehow model abstractly the “place holder” where all voting “goes”.

Further, we have identified the following Rs:

AR1-AR2 suggesting that the CAMPAIGNER is promoting political messages that are supposed to influence the VOTER;

AR2-SYSTEM suggesting that the VOTER provides essential input to the SYSTEM, namely the vote;

SYSTEM-AR3 suggesting that the SYSTEM has impact with regard to each voting station (said otherwise, each voting station has its “own” SYSTEM), by providing the information needed by

the PRIMARY COUNTER for calculating the station results;

AR3-AR4 suggesting that the SECONDARY COUNTER needs the PRIMARY COUNTER’s feedback from each voting station, in order to aggregate the overall voting results;

AR5-ALL suggesting the ORGANIZER of the elections has relationship with all above-mentioned ARs and the SYSTEM as follows: creating conditions for the CAMPAIGNER to do promotion adequately; establishing that the rights of the VOTER are guaranteed; establishing rules and mechanisms according to which the PRIMARY COUNTER and the SECONDARY COUNTER should fulfil their corresponding tasks; establishing and running the voting SYSTEM;

A6-ALL suggesting that the CONTROLLER should execute effective control concerning all above-mentioned ARs and the SYSTEM, as guarantee that the voting is fair.

This is the basis for our conceptual model and as mentioned before, we abstract from several issues (as according to Sub-Section 3.2) and an example of this is that we do not consider an AR pointing to the one(s) (outside the CAMPAIGNER) who may be somehow influencing the decision of the VOTER – this could have been modeled as an AR by itself but we have not done this because of the lack of technical relevance, as explained above.

We present our conceptual model on Figure 1 and we use simple and intuitive graphical notations: the labels of the ARs are put inside boxes and the SYSTEM is presented as oval, while the Rs are represented as lines (the arrows indicate who is ADDRESSED in the relationship – for example: if the CAMPAIGNER is influencing the VOTER, then the arrow should be at the VOTER end because the VOTER is addressed by this).

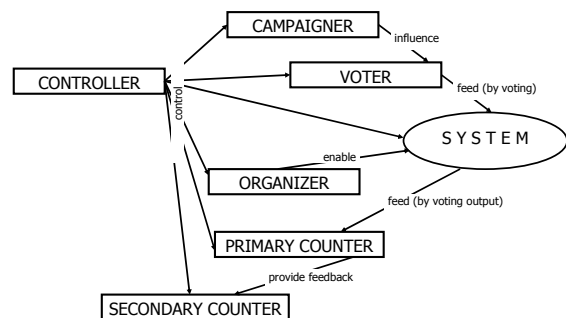


Figure 1: The Voting Conceptual Model.

As seen on the figure, we have not only drawn arrows at each line (lines representing Rs) but we

have also added labels there: the CAMPAIGNER would influence the VOTER, the ORGANIZER would enable the SYSTEM, and so on.

We claim that such an essential conceptual model is a good basis for further elaborations in different directions, such as structure (where we model entities and their relations), behaviour (where we model all activities in sequence, parallelism, and so on), data, and so on. The model would stay as “guarantee” for the inter-model consistency among structural, behavioural, data, and other “sub”-models. For the sake of brevity, we are not going in more details in presenting and discussing those issues. Still, the SDBC approach provides useful insight in this direction (Shishkov, 2005).

In line with the goals of the current paper, as mentioned in the Introduction, we go for extending the conceptual model, by identifying (technology-inspired) requirements, as part of the effort of supporting the development of eVoting systems. This will be addressed in the next section.

4 eVOTING REQUIREMENTS

In the current section, we will firstly elaborate the public demands (see Sub-Section 3.1) and then we will introduce our way of modeling requirements, being certainly restricted by the conceptual model (see Sub-Section 3.3).

4.1 Public Demands' Elaboration

In the current sub-section, we elaborate the previously listed public demands towards eVoting, taking into account that all those issues concern the people and the technology (what the current technological possibilities are), and the legislation. The current demands' elaboration would be useful as basis for our reflecting the demands in corresponding technical requirements.

With regard to the SECRECY OF VOTE demand, there are two things: (i) it is to be guaranteed that **nobody can know how a person has voted**; (ii) it is to be ensured that the person has been marked as “voted”, such that (s)he **would not go to vote again**.

With regard to the COST ADEQUACY demand, the only way of avoiding the “big expensive black box” is to conceptualize the eVoting process such that **it is known what technology is needed for what**.

A way to guarantee against VIOLATIONS with regard to the way the System is working, is to present

the user with a **simple and exhaustive list of options**, with no possibilities to do anything outside the presented options.

A way to guarantee against MANIPULATIONS OF THE FINAL RESULTS is to keep things at two levels, such that the **Primary Counters generate the “raw” results** based on which the **Secondary Counters generate the final results** and this all stays stored with **possibility to check in the future**.

The COMPUTER-SERVER communication is to be such that there is guarantee that a **“packet” sent by a computer is received by the server and by noone else**; this is a matter of organization and also a matter of networking protocols.

CONTROLLABILITY can be partially achieved if **all intermediary results get transparent** and then the only remaining challenge is how are the “raw” results generated.

FAULT REACTION is a matter of **recoverability** and this is a non-functional concern that has to be addressed from a functional perspective nevertheless.

EASE OF USE is a matter of design.

The issue on QUALIFICATIONS needed for being involved in eVoting is a matter of legislation; as it was mentioned before, sufficient IT literacy among the population is assumed.

4.2 Way of Modeling Requirements

We consider OS norms (see Sub-Section 2.2) as helpful in the process of specifying requirements in this context not only because Organizational Semiotics is well-known for its strengths with regard to capturing societal aspects but also because OS norms have been researched also as useful with regard to requirements identification and specification. In particular, it has been studied how OS norms can help deriving use cases from business processes (Shishkov, 2005).

OS norms determine the conditions and constrains in controlling optional and conditional actions. They govern the behaviour of actors (agents), normally to decide when certain actions are performed. OS norms define clearly the roles, functions, responsibilities and authorities of the actors, for example:

```

whenever John is Customer of VISA
if VISA increase John's credit card limit
then John
is allowed
to use more credit.

```

4.3 The eVoting Requirements

Starting from unstructured information – the public demands presented in Sub-Section 3.1 (a mixture of technology-independent and technical issues), we had to firstly capture the functional gist (the conceptual model), abstracting from technical details – see Figure 1. Only on this basis, it was possible to adequately elaborate the public demands (see Sub-Section 4.1), making sure that each (technical) detail properly fits in the “big picture”.

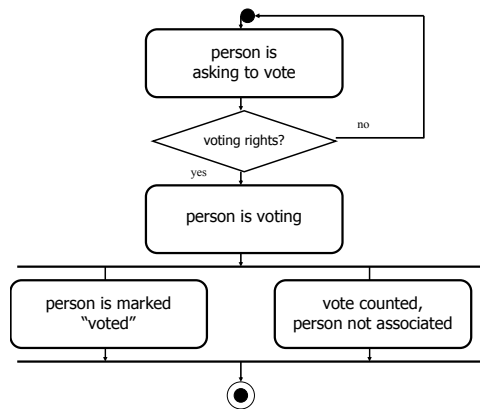


Figure 2: Workflow Pattern Corresponding to OS Norm 1.

Hence, inspired by Organizational Semiotics, we can now specify eVoting requirements, by means of OS norms, and in order to achieve better visualization and possible good basis for simulation (it this may be needed), we also reflect the OS norms in workflow, as studied by Shishkov (2005). Further, the demands being considered are many while the scope of the current paper is limited; hence, for the sake of brevity, we take only several of them (planning the rest as future work) and reflect them in the specification of requirements and in the next section, we briefly discuss the next step: “requirements-to-architecture”.

We take in this section several eVoting public demands and we reflect them in specified requirements expressed in terms of OS norms. We start from the SECRECY OF VOTE one:

OS Norm 1:

Whenever John has voting rights
 if John is executing eVoting
 then the eVoting system
 is (i) obliged to mark John as “voted”
 is (ii) prohibited from recording the way John has voted.

Based on OS Norm 1, we derive a workflow pattern expressed with the notations of UML Activity Diagram (UML, 2016) – see Fig. 2.

With regard to the VIOLATION public demand, we formulate the following norm:

OS Norm 2:

whenever John is executing eVoting
 if John is attempting a not allowed action
 then the eVoting system
 is prohibited from taking any action.

Based on OS Norm 2, we derive a workflow pattern – see Fig. 3.

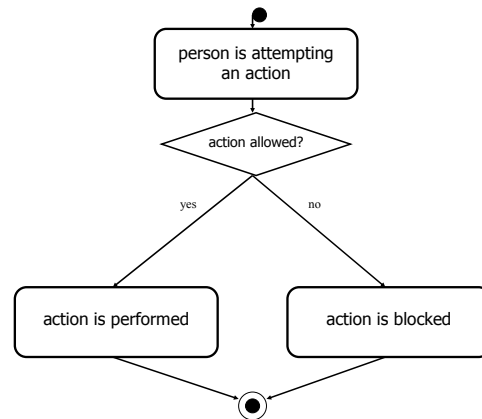


Figure 3: Workflow Pattern Corresponding to OS Norm 2.

5 TOWARDS A SERVICE-ORIENTED ARCHITECTURE

With regard to the “requirements-to-architecture” step, we consider the SDBC Approach (Shishkov, 2005) allowing for a component-based alignment between enterprise modeling and software design, ending up in specified software components that fit in the architecture. Those components we relate to web services in a way that what we see as web service “manifestation” is the functionality delivered by corresponding underlying software component(s). Further, we realize that in currently dealing with distributed cloud applications, it would often be that different software components have different origin thus not belonging to the same software application. Still, the SDBC Architecture is that needed abstraction which establishes and keeps the overall system “logic” no matter if a particular task is realized by the software application –to-be or by (dynamically) composed web services. Further detailing is left for future research. What we present in the current section is the SDBC-orientation we take with regard to architecture, noting the useful relation

to web services. For this reason, we briefly present below the SDBC Approach, SDBC standing for: ‘Software Derived from Business Components’.

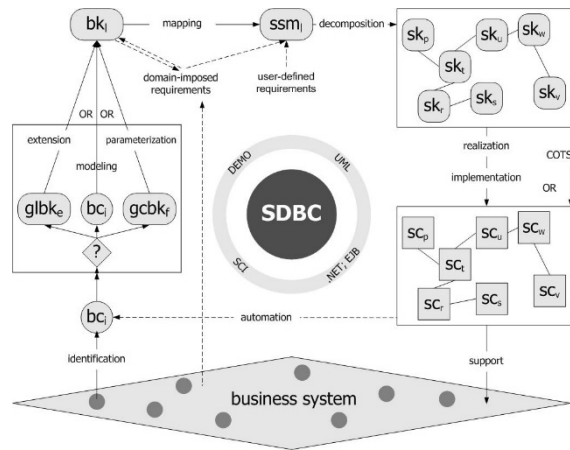
Firstly, SDBC assumes 4 modeling perspectives, namely: *Structural Perspective* that reflects entities and their relationships; *Dynamic Perspective* that reflects the overall business process and corresponding to this – the states of each entity, evolving accordingly; *Data Perspective* that reflects the information flows across entities and within the business process; *Language-Action Perspective* that reflects real-life human communication and expression of promises, commitments, etc. as also relevant to soundly building an exhaustive enterprise model.

Secondly, among SDBC’s underlying theories are Enterprise Ontology and Organizational Semiotics (Shishkov et al., 2006) - see Section 2, which makes the approach suitable especially in the eVoting context of the current paper.

Thirdly, among the main SDBC concepts are the following:

- **Component vs CoMponent:** while components represent part of the whole, coMponents reflect a model of a component adequately elaborated in all 4 perspectives (see above), and we could thus have business components (business sub-systems) and software components (pieces of implemented software) as well as business coMponents and software coMponents, respectively;
- **General vs Generic:** those concepts are both about re-use, still – general is about re-using an abstract core (a general reservation engine, for example) while generic is about parameterizing something that is multi-specific (a car system to be adjusted to automatic or gear regime, for example);
- **Software Specification Model** – this is a technology-independent functionality model of the software system-to-be.

To summarize the SDBC outline, we use Fig. 4:



Abbreviations:

- | | |
|-----------------------------------|------------------------------------|
| bc – Business Component | ssm – Software specification model |
| bk – Business CoMponent | sc – Software Component |
| glbk – General Business CoMponent | sk – Software CoMponent |
| gcbk – Generic Business CoMponent | |

Figure 4: SDBC – Outline (Shishkov, 2005).

As seen from the figure, we consider a Business System from which a Business Component(s) is to be identified and then reflected in a relevant model – a Business CoMponent. Another way for arriving at a Business CoMponent is by applying re-use: either extending a general Business CoMponent or parameterizing a generic one. Then, the Business CoMponent should be elaborated with the **domain-imposed requirements**, in order to add elicitation on the particular context in which its corresponding Business Component exists within the Business System. Then, a mapping towards a software specification model should take place and the **user-defined requirements** are to be considered, since the derived software model should reflect not only the original business features but also the particular requirements towards the software system-to-be. The software specification model in turn needs a precise elaboration so that it provides sufficient elicitation in terms of structure, dynamics, data and language-action –related aspects. It needs also to be decomposed into a number of Software CoMponents reflecting functionality pieces. Those CoMponents then are to undergo realization and implementation, being reflected in this way in a set of Software Components. Some Software Components could also be purchased. The Software Components are implemented using Software Component technologies, such as .NET or EJB, for instance. Finally, the (resulting) component-based ICT application would support informationally the target Business System, by automating anything that

concerns the considered Business Component (identified from the mentioned system).

SDBC was just briefly introduced above. Still, interested readers can find more information on SDBC in (Shishkov, 2005).

As mentioned in the previous section, we just briefly discuss our views on the “requirements-to-architecture” step in the eVoting context, and we leave it for future research to go in depth in this direction, possibly adapting some SDBC features accordingly, and of course validating our results by means of case studies. Still, it becomes clear how we proceed from requirements to architecture and how we consider service-orientation.

6 CONCLUSIONS

Understanding eVoting systems requires understanding the legislative and societal context. In this position paper we presented guidelines to come from global to detailed requirements, and then – to (service-oriented) architecture, based on Enterprise Ontology, Organizational Semiotics, and the SDBC Approach. The essence is that multiple theories need to be employed to understand and elicit requirements. Formal modeling should ensure that the requirements are consistent and meet the societal expectations.

REFERENCES

- Dietz, J.L.G., 2006. *Enterprise Ontology, Theory and Methodology*. Springer-Verlag, Berlin Heidelberg.
- Insfran, E., Pastor, O., Wieringa, R., 2002. Requirements Engineering-Based Conceptual Modelling. In *Requirements Engineering journal*, Vol. 7, Nr. 2, 2002. Springer-Verlag.
- Konstantinov, M., Pelova, G., Boneva, J., 2009. Mathematics of the Bulgarian Electoral System. In *AMEE'09, 35th International Conference on Applications of Mathematics in Engineering and Economics*. AIP.
- Liu, K., 2000. *Semiotics in Information Systems Engineering*. Cambridge University Press, Cambridge.
- Papazoglou, M., 2012. *Web Services and SOA: Principles and Technology*, Prentice Hall, 2nd edition.
- Plevneliev, R., 2016. *Statement by President Rosen Plevneliev at His 4th Annual Press Conference*, Published on the website of the President of the Republic of Bulgaria: <http://www.president.bg>.
- Scammell, R., 2016. *Internet Voting a Success in Two European Countries*, Published on the European University Institute website: <http://www.eui.eu>.
- Shishkov, B., 2005. *Software Specification Based on Reusable Business Components (PhD Thesis)*, TU Delft – SIKS Publishing, Delft.
- Shishkov, B., Dietz, J.L.G., Liu, K., 2006. Bridging the Language-Action Perspective and Organizational Semiotics in SDBC. In *ICEIS'06, 8th International Conference on Enterprise Information Systems*. SCITEPRESS.
- UML, 2016, the website on the Unified Modeling Language: <http://www.uml.org>.
- Van der Aalst, W., 2011. *Process Mining - Discovery, Conformance and Enhancement of Business Processes*. Springer-Verlag, Berlin Heidelberg.

A Framework for Process Driven Software Configuration

Andreas Daniel Sinnhofer¹, Peter Pühringer, Klaus Potzmader², Clemens Orthacker²,
Christian Steger¹ and Christian Kreiner¹

¹*Institute of Technical Informatics, Graz University of Technology, Graz, Austria*

²*NXP Semiconductors, Gratkorn, Austria*

{a.sinnhofer, christian.kreiner, steger}@tugraz.at, p.puehringer@inode.at, {klaus.potzmader, clemens.orthacker}@nxp.com

Keywords: Software Product Lines, Feature Oriented Modelling, Business Processes, Tool Configuration.

Abstract: Business processes have proven to be essential for organisations to be highly flexible and competitive in today's markets. However, good process management is not enough to survive in a market if the according IT landscape is not aligned to the business processes. Especially industries focused on software products are facing big problems if the according processes are not aligned to the overall software system architecture. Often, a lot of development resources are spent for features which are never addressed by any business goals, leading to unnecessary development costs. In this paper, a framework for a business process driven software product line configuration will be presented, to provide a systematic way to configure software toolchains.

1 INTRODUCTION

Business Process (BP) oriented organisations are known to perform better regarding highly flexible demands of the market and fast production cycles (e.g. McCormack and Johnson (2000); Valena et al. (2013); Willaert et al. (2007)). This is achieved through the introduction of a management process, where business processes are modelled, analysed and optimised in iterative ways. Nowadays, business process management is also coupled with a workflow management, providing the ability to integrate the responsible participants into the process and to monitor the correct execution of it in each process step. To administer the rising requirements, so called business process management tools are used (BPM-Tools) which cover process modelling, optimization and execution. In combination with an Enterprise-Resource-Planning (ERP) system, the data of the real process can be integrated into the management process.

In the domain of software products, different choices in business processes lead to different software configurations. To handle variability automatically is a challenging task because the variability of the process model needs to be reflected in the software architecture. Further, the actual customer choice during the ordering process needs to be mapped to the according software features. Due to this, software configuration is often done manually which takes a considerable amount of time during production. Partic-

ularly for resource constraint devices like embedded systems, it is vital to have a working software configuration process since unnecessary features may occupy a lot of memory. Further, it is important to have a software architecture which is synchronised with the business goals. Otherwise, a lot of resources are spent for developing and maintaining software components which are never used anyway. Thus, process awareness is crucial for an efficient development.

Context Aware Business Process modelling is a technique for businesses living in a complex and dynamic environment (Saidani and Nurcan (2007)). In such an environment a company needs to tackle changing requirements which are dependent on the context of the system. Such context sensitive business process models are able to adapt the execution of their process instances according to the needs, such that the company can react faster and more flexible. This is achieved by analysing the context states of the environment and mapping these states to the according business processes and their related software system. The problem with such approaches is, that the used software systems are often developed independently from each other, although they share a similar software architecture. Therefore, this work focuses on the development of a framework which covers the variability of process models and mapping such variable process structures to software configuration artefacts such that the software system can be adapted automatically with respect to its context. This

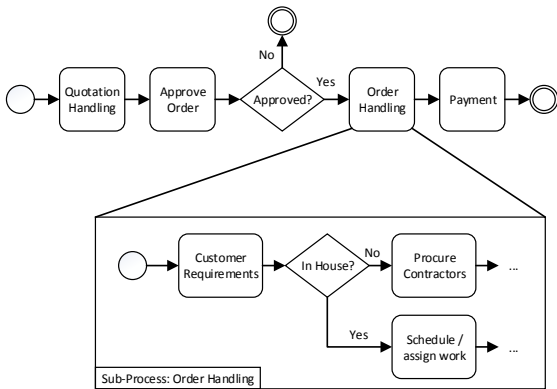


Figure 1: Exemplary order process to illustrate the basic concepts defined by Österle (1995): A high level description of the process is split into its sub-processes until a complete work description is reached.

is achieved through software product line engineering techniques. Thus, only one system needs to be developed and maintained for whole product families. The modelling of business process variability is based on our previous work, which can be found in Sinnhofer et al. (2015). In particular, a SPLE Tool was used to systematically reuse expert knowledge in form of valid process variations, designed in an appropriated BPM Tool. The integrity of the process variations is secured by the capabilities of the BPM Tool and a rich cross functional constraint checking in the SPLE Tool. This work will extend the framework in order to be able to map process artefacts to software configurations. Hence, software toolchains can be configured in an automatic way and the architecture can be kept aligned with the business goals.

This work is structured in the following way: Section 2 gives an overview over the used design paradigm for business processes modelling and Software Product Line Engineering techniques which were needed for the framework. Section 3 summarizes the concept of our work and Section 4 describes our implementation in an industrial use case. Finally, Section 5 summarizes the related work and Section 6 concludes this work and gives an overview over future work.

2 BACKGROUND

2.1 Business Processes

A business process can be seen as a sequence of tasks/sub-processes which need to be executed in a specific way to produce a specific output with value to the customer (Hammer and Champy (1993)). Ac-

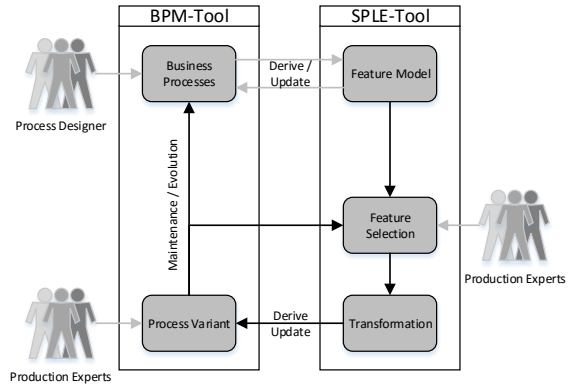


Figure 2: Used framework for an automatic business process variant generation (adapted from Sinnhofer et al. (2015)). The grey lines indicate process steps which need to be done manually.

ording to Österle (1995) the process design on a macroscopic level (high degree of abstraction) is split up into sub-processes until the microscopic level is reached. This is achieved, when all tasks are detailed enough, so that they can be used as work instructions. An exemplary order process is illustrated in Figure 1. As illustrated, the top layer is a highly abstracted description, while the production steps are further refined on the lower levels. As a result, the lowest level is highly dependable on the concrete product and production environment, providing many details for the employees. Usually, the top layers are independent from the concrete plant and the supply chain and could be interchanged throughout production plants. Only the lower levels (the refinements) would need to be reconsidered. Variability of such a process structure can either be expressed through a variable structure of a process/sub-process (e.g. adding/removing nodes in a sequence) or by replacing the process refinement with different processes.

Traditionally, processes for similar products are created using a copy and clone strategy. As a result, maintaining such similar processes is a time consuming task, since every improvement needs to be propagated manually to the respective processes. To solve this issue, we proposed a framework to automatically derive process variants from business process models by modelling the variable parts of a process using Software Product Line Engineering techniques in a previous work (see Sinnhofer et al. (2015)). The presented framework can be split into four different phases which are illustrated in Figure 2. In the first phase, process designers create process templates in a BPM tool, adding all wished features like documentation artefacts, responsible workers or resources. In the second phase, the created processes are imported into the SPLE tool and added to a feature model. Pro-

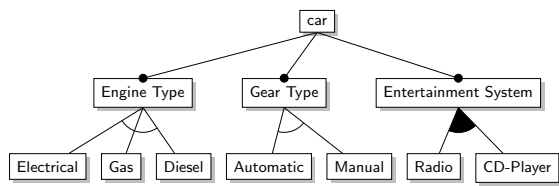


Figure 3: An exemplary feature model of a car.

cess experts define a comprehensive set of rules and restrictions so that only valid process variants can be derived from the model. The third phase is called the feature selection phase in which production experts will automatically derive processes for their needs based on a selection of features. The fourth phase consists of maintenance and evolution. There, data is collected and used to improve process designs or feature selections.

2.2 Software Product Line Engineering

SPLE applies the concepts of product lines to software products (Kang et al. (1990)). A Software Product Line can be seen as a set of domain features, which are automatically assembled and configured to a whole software project just by choosing the wanted features. Instead of writing code for a whole system, the developer divides the project into small lightweight features which are implemented in so called domain artefacts. For this, a software architecture is needed in which the variation points and the respective variants (features) are explicitly modelled. Further, a source code generator is needed which is able to generate the according software products, based on the according feature selection.

Features are usually modelled in so called 'Feature Models' which describe all features of a product and explicitly states their relationships, dependencies and additional restrictions between each other. Figure 3 illustrates an explanatory feature model for a car. A car consists of three mandatory variation points (Engine Type, Gear Type, Entertainment System) and their respective variants. For example, the Engine Type of the car could be Electrical, Gas or Diesel powered. The variants of the 'Engine Type' and 'Gear Type' variation point are modelled as alternative features which means that exactly one variant needs to be chosen. In contrast, the 'Entertainment System' is modelled in such a way, that either one or both options can be chosen.

3 VARIABILITY FRAMEWORK

The goal of the developed framework is to implement

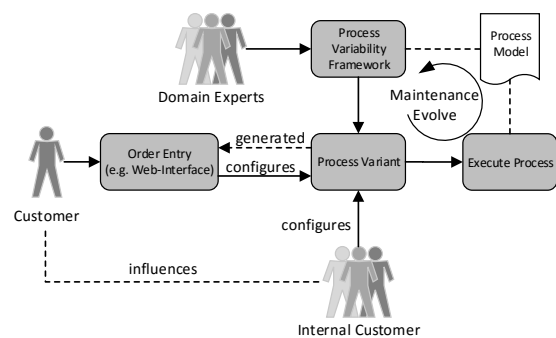


Figure 4: Overall conceptual design of the framework. The "Process Variability Framework" block is described in Figure 2.

a systematic way to keep the business processes aligned with the IT infrastructure so that development costs can be reduced and the company is more flexible to changes of the market. The following Sections summarizes our developed framework.

3.1 Conceptual Design

The overall conceptual design is based on a feature oriented domain modelling framework and is displayed in Figure 4. As illustrated in the Figure, Domain Experts are responsible for operating the "Process Variability Framework" as already described in Section 2.1. They design process models based on their domain knowledge and generate process variants for various types of product platforms. Based on this variants, the used SPLE tool also generates a order entry form, stating explicitly which kind of information a customer needs to submit, to be able to order the product. For example, if the customer can decide which applications should run on his device or if the device can be personalized by adding signatures of the customer. Complex products usually tend to have a lot of internal stakeholders which can be seen as internal customers. This means that based on the customer needs, specific stackholders may be addressed to further submit needed information or even parts of the product. For instance, if a product can run on multiple hardware platforms, each of these platforms may be developed by different departments or even different companies which need to be ordered and shipped accordingly. To be able to automatically generate the order entry forms, additional information needs to be added to the process models. This can be done by either adding this information into the process model itself (i.e. using the BPM tool) or by using the capabilities of the SPLE tool and mapping this information to the according process models. Option two is the more generic approach which also has the

positive side-effect, that the processes itself are not "polluted" with information that may change in different circumstances. On the other hand, it rises high requirements to the SPLE tool which needs to support product family models so that the process model and the additional information used for the order entry can be kept aligned, but separated which increases the reusability factor.

After all needed data is collected, the process can finally be executed and the ordered products are manufactured. Especially for new products, it is likely that during this manufacturing process knowledge is gained on how to increase the efficiency of the whole process(es) by introducing specific changes to the process model. Further, changes to the generated order entry may be identified, which means that specific parts of the product need to be made selectable. The advantage of using one core of process models for a specific family of products is that the gained knowledge can be rolled out in an automatic way for the whole product family. This means that the required changes only need to be implemented once.

3.2 Type Model

To automatically generate order entry forms from a feature selection, the used model needs to support the following types:

- **Inputs:** Is the abstract concept of different Input types which are described below.
- **None:** No special data needs to be submitted and hence a node (i.e. task in a process) marked with none will not appear as a setting in the order entry form.
- **Customer Input:** Specific data need to be added from a customer. A node marked with this will generate an entry in the order entry form of a specific type. For example a file upload button will appear if a customer needs to submit specific files.
- **Internal Input:** Specific data or parts of the product needs to be delivered from an internal stackholder. This information is directly submitted to the internal stackholder as a separate order.

Furthermore, the family model should support the concept of choices (i.e. a customer needs to submit one of possible n options) and multiple inputs if multiple submissions are needed for a single node. Also multiple inputs of multiple different stackholder need to be supported.

3.3 Process Driven Software Toolchain Configuration

An established process management, which is able to generate order entry forms and trigger internal processes, is a big step towards good business management. However, to be successful on the market it is not enough to just focus on well managed processes, but also on an aligned IT infrastructure. Hence, the big remaining challenge is having an IT infrastructure which is able to be configured directly from the according business processes.

For illustration purposes let's consider the following example: A company is developing small embedded systems which are used as sensing devices for the internet of things. The device is offered in three different variants with the following features:

- **Version 1:** Senses the data in a given time interval and sends the recorder signal to a web-server which is used for post-processing.
- **Version 2:** Additionally to the features of Version 1, this version allows encryption of the sensed data using symmetric cryptography before it is sent to the web-server. This prevents that third parties are able to read the data. For simplicity, we assume that this key is provided in plain from the customer.
- **Version 3:** Additionally to the features of Version 2, this version also allows customer applications to be run (e.g. data pre-processing routines) on the system.

It is not economic feasible to personalize each device manually if it is sold in high quantities. Further, establishing three different order processes using three different versions of customization toolchains will result in higher maintenance efforts. To summarize the findings of this short example, it is fundamental to have a software architecture which is synchronized with the according business process(es). This means that variable parts of the process model need to be reflected by a variable software architecture. Further, minor changes to the process model (e.g. addition of new configuration settings) should not lead to huge development efforts since – ideally – the software architecture does not need to be changed. These requirements lead to the architecture displayed in Figure 5. As illustrated, the tool is basically an interpreter which can be "dynamically programmed" for the actual order. This means that variability of the architecture is gained by shifting features from the implementation phase to the configuration phase. To ensure that such freedom is not misused, it is necessary to enforce specific rules in the Interpreter Tool (e.g.

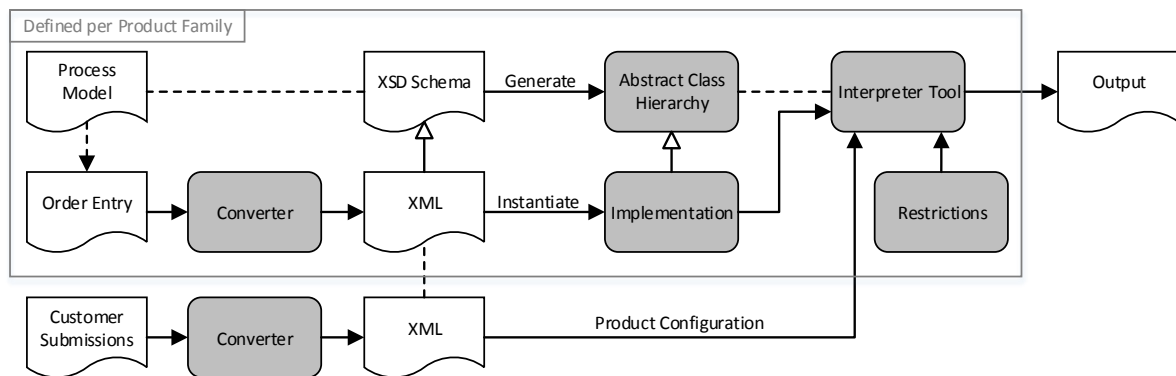


Figure 5: The architecture of the software tool responsible for generating the wished product outcome.

security requirements). Based on the Process Model of the Process Variability Framework, a schema file is created which states all possible operations and all additional language primitives (like conditional statements, etc.) the Interpreter Tool can perform. This step is semi-automatic which means that only a skeleton of the needed functionality can be generated automatically.

For illustration purposes we will reconsider the previous example: Basically, there are three different order processes, where in the first case a customer can customize a connection string for his web-server. In the second case he can further submit a key which is stored onto the nodes and in the third case executables can be submitted to be loaded to the chip. Taking this into account, the XML illustrated in Listing 1 can be generated. Each function consists of a *Configuration* block and a *Translate* block. The Configuration block is used to indicate which data needs to be provided from the customer submissions (i.e. from the "real"

product configuration) and how often they can occur (configuration safety). This Configuration blocks are further used to generate a schema file which is used by the converter tool to convert the Customer Submissions into the needed XML structure. The Translate block defines how the submitted data is processed. This cannot be generated and hence a developer is needed who needs to define this transformation based on the language primitives of the Interpreter Tool. This needs to be done only once for a whole product family. For this particular example, only a Store-Data and an Install-Application routine would need to be offered by the interpreter. Additional restrictions are domain depended and could contain in that example the following checks: Verification that the submitted key is of reasonable strength (e.g. AES key with a minimum length of 16 bit) and that the submitted applications are protected by a signature of the customer to ensure that they are not replaced by a malicious third party. If a product is ordered, the filled order entry (i.e. customer submissions) is converted into a

Listing 1: Generated XML based on the Order Entry. The Translate blocks need to be edited manually by a developer.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <Functions>
3    <Function id="WebServer"
4      minOccurs="1"
5      maxOccurs="1">
6      <Configuration>
7        <Parameter name="Connection" type="ipAddress" />
8      </Configuration>
9      <Translate> ... </Translate>
10   </Function>
11   <Function id="EncryptionKey"
12     minOccurs="0"
13     maxOccurs="1">
14     <Configuration>
15       <Parameter name="Key" type="hexstring" />
16     </Configuration>
17     <Translate> ... </Translate>
18   </Function>
19   <Function id="InstallApplication"
20     minOccurs="0"
21     maxOccurs="unbounded">
22     <Configuration>
23       <Parameter name="Binary" type="fileUri" />
24     </Configuration>
25     <Translate> ... </Translate>
26   </Function>
27 </Functions>

```

Listing 2: Exemplary generated Configuration file based on customer submissions. Two different versions are shown. The first example illustrates a "Version 3" product and the second one a "Version 1" product.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <CustomerOrder>
3    <WebServer>
4      <Connection>X.X.X.X</Connection>
5    </WebServer>
6    <EncryptionKey>
7      <Key>0x01020304 ... </Key>
8    </EncryptionKey>
9    <InstallApplication>
10     <Binary>file://orderXYZ/app1.elf</Binary>
11   </InstallApplication>
12   <InstallApplication>
13     <Binary>file://orderXYZ/app2.elf</Binary>
14   </InstallApplication>
15 </CustomerOrder>

1  <?xml version="1.0" encoding="UTF-8"?>
2  <CustomerOrder>
3    <WebServer>
4      <Connection>X.X.X.X</Connection>
5    </WebServer>
6 </CustomerOrder>

```

configuration file which instantiates the specific features of the product. For example Listing 2 shows the generated Configuration file for a "Version 3" product (top one) and a "Version 1" product (bottom one).

4 INDUSTRIAL CASE STUDY

In this section an overview over our industrial case study is given. The implemented business processes of our industrial partner are controlled by an SAP infrastructure and are designed with the BPM-Tool Aeneis. Further, pure::variants is used as SPLE tool to manage the variability of the business processes. Thus, our implemented prototype is also based on pure::variants and Java.

4.1 SPLE-Tool: pure::variants

pure::variants is a feature oriented domain modelling tool which is based on Eclipse. As such, it can easily be extended based on Java plug-in development. During the implementation of this project, five different plug-ins were developed:

- An extension to the import plug-in which was developed in our previous work. It assists the Process Designers in modelling cross functional requirements and providing the needed information for the code generators.
- An extension to the internal model compare engine for comparing different versions of created feature models with each other.
- An extension to the internal model transformation engine to convert the feature selection of the process model into the according order entry form. This also generates the back-end to trigger processes for internal stakeholders.
- Additions to the internal model check engine to model and create only valid processes (e.g. checks related to the feature selection, the consistency of the feature model, etc.)
- Generator Tools which are able to generate the skeleton of the schema file (as described in Section 3.3) and the order entry form (a generated Web-Interface). Additionally, converter tools were written which are converting the generated forms and received submissions into the related XML files.

4.2 Implementation of the Interpreter Tool

As mentioned in Section 3.3, the class hierarchy should be generated from a schema file, thus we used the tool Jaxb (a Java architecture for XML binding) to generate the bare class hierarchy which needs to be implemented by the software developers. Since the creation of the schema file is semi-automatic, our developed framework (implemented in pure::variants) opens a dialogue which hints the domain expert to check the validity of the schema file to ensure that the changes to the processes are always propagated to the schema file. Since our industry partner is working in a safety and security critical domain, additional restrictions are implemented. Formal verification rules are implemented to check that confidential data is not leaked and rules are defined to check the configuration safety such that no invalid configuration can be submitted and executed. These restrictions will be part of a future publication.

4.3 Evaluation

The framework was successfully deployed for two different product families which are based on the same Process Model. The time was measured to implement the initial system and the overhead to support the two systems to get an effort estimation which can be compared with a traditional software development. We use the term "traditional software development" for a software development with ad-hoc (or near to ad-hoc) software architecture which means that multiple different systems are designed almost independently. This leads to the situation that only a little code base is shared between each software project since most of the code is optimized for single purposes. However, this code would be reusable if adaptations of the interfaces / implementations would have been considered. The effort for the traditional software development was based on the real implementation time for the first system and an effort estimation to port the existing family to the new one. These numbers were given by the responsible developers. As illustrated in Table 1, the break-even point will be between 3 to 4

Table 1: Effort measurements and estimations in man-month to develop the systems.

	Framework	Traditional
Base System	12	-
Product Fam. 1	1	6
Product Fam. 2	0.5	4 - 5
Overall	14,5	10 - 11

systems using a curve fitting interpolation. This number also correlates to the typical number presented in relevant software product line publications (e.g. Pohl et al. (2005)). Additionally, the maintenance cost can be reduced since fixing problems in one product family will fix this issue in all others as well.

5 RELATED WORK

As stated in the survey of Fantinato et al. (2012), major challenges in the field of business process variability modelling are related to the reaction time of process changes and of the creation and selection of the right business process variants, which are also main topics in our framework since the time to adopt the IT infrastructure to the changed business processes can be reduced with the new framework.

Derguech (2010) presents a framework for the systematic reuse of process models. In contrast to this work, it captures the variability of the process model at the business goal level and describes how to integrate new goals/sub-goals into the existing data structure. The variability of the process is not addressed in his work.

Gimenes et al. (2008) presents a feature based approach to support e-contract negotiation based on web-services (WS). A meta-model for WS-contract representation is given and a way is shown how to integrate the variability of these contracts into the business processes to enable a process automation. It does not address the variability of the process itself but enables the ability to reuse business processes for different e-contract negotiations.

While our used framework to model process variability reduces the overall process complexity by splitting up the process into layers with increasing details, the PROVOP project (Hallerbach et al. (2009a,b) and Reichert et al. (2014)) focuses on the concept, that variants are derived from a basic process definition through well-defined change operations (ranging from the deletion, addition, moving of model elements or the adaptation of an element attribute). In fact, the basic process expresses all possible variants at once, leading to a big process model. Their approach could be beneficial considering that cross functional requirements can be located in a single process description, but having one huge process is also contra productive (e.g. the exchange of parts of the process is difficult).

The work Gottschalk et al. (2007) presents an approach for the automated configuration of workflow models within a workflow modelling language. The term workflow model is used for the specification

of a business process which enables the execution in an enterprise and workflow management system. The approach focuses on the activation or deactivation of actions and thus is comparable to the PROVOP project for the workflow model domain.

Rosa et al. (2008) extends the configurable process modelling notation developed from Gottschalk et al. (2007) with notions of roles and objects providing a way to address not only the variability of the control-flow of a workflow model but also of the related resources and responsibilities.

The Common Variability Language (CVL Haugen et al. (2013)) is a language for specifying and resolving variability independent from the domain of the application. It facilitates the specification and resolution of variability over any instance of any language defined using a MOF-based meta-model. A CVL based variability modelling and a BPM model with an appropriate model transformation could lead to similar results as presented in this paper.

The work of Zhao and Zou (2011) shows a framework for the generation of software modules based on business processes. They use clustering algorithms to analyse dependencies among data and tasks, captured in business processes. Further, they group the strongly dependent tasks and data into a software component.

6 CONCLUSION AND OUTLOOK

The reuse of business process models is an important step for an industrial company to survive in a competitive market. But only with an integrated view of the according IT landscape it is possible to raise the efficiency of the overall business. With this work we proposed a way to combine the benefits of software product line engineering techniques with the capabilities of a business process modelling tool. This work provides a framework for the systematic reuse of business processes and the configuration of software toolchains used during the actual production of the product. The new introduced framework is able to synchronize variable process structures with a variable software architecture. This means that changes to the processes will automatically generate a skeleton of the software artefacts which need to be implemented by the developers. For that, the framework uses XML data binding to bind specific software features to a specific set of configurable artefacts which need to be submitted by customers (internal and external) during the order process. This is done in an automatic and managed way so that the order interface is always aligned to the software toolchains. More-

over, the overall robustness of the software toolchains is increased since the same code base is shared for a lot of different product families leading to a higher customer satisfaction.

Future work will address the semi-automatic creation of the schema file which is used to keep the software architecture aligned to the process models. Another point for improvement is the fact that additional security requirements are implemented and mapped manually to the according product configurations. In a future work, we will investigate a way to map these security requirements to the according process model which enables an automatic way to bind these requirements to the product families and enforce them in the process. This is important especially if a certification of the products is intended.

ACKNOWLEDGEMENT

The project is funded by the Austrian Research Promotion Agency (FFG). Project Partners are NXP Semiconductor Austria GmbH and the Technical University of Graz. We want to gratefully thank Danilo Beuche from pure::systems for his support.

REFERENCES

- Derguech, W. (2010). Towards a Framework for Business Process Models Reuse. In *The CAiSE Doctoral Consortium*.
- Fantinato, M., Toledo, M. B. F. d., Thom, L. H., Gimenes, I. M. d. S., Rocha, R. d. S., and Garcia, D. Z. G. (2012). A survey on reuse in the business process management domain. *International Journal of Business Process Integration and Management*.
- Gimenes, I., Fantinato, M., and Toledo, M. (2008). A Product Line for Business Process Management. *Software Product Line Conference, International*, pages 265–274.
- Gottschalk, F., van der Aalst, W. M. P., Jansen-Vullers, M. H., and Rosa, M. L. (2007). Configurable Workflow Models. *International Journal of Cooperative Information Systems*.
- Hallerbach, A., Bauer, T., and Reichert, M. (2009a). Guaranteeing Soundness of Configurable Process Variants in Provop. In *Commerce and Enterprise Computing, 2009. CEC '09. IEEE Conference on*, pages 98–105. IEEE.
- Hallerbach, A., Bauer, T., and Reichert, M. (2009b). Issues in modeling process variants with Provop. In Ardagna, D., Mecella, M., and Yang, J., editors, *Business Process Management Workshops*, volume 17 of *Lecture Notes in Business Information Processing*, pages 56–67. Springer Berlin Heidelberg.
- Hammer, M. and Champy, J. (1993). *Reengineering the Corporation - A Manifesto For Business Revolution*. Harper Business.
- Haugen, O., Wasowski, A., and Czarnecki, K. (2013). Cvl: Common variability language. In *Proceedings of the 17th International Software Product Line Conference, SPLC '13*.
- Kang, K., Cohen, S., Hess, J., Novak, W., and Peterson, A. (1990). Feature-oriented domain analysis (foda) feasibility study.
- McCormack, K. P. and Johnson, W. C. (2000). *Business Process Orientation: Gaining the E-Business Competitive Advantage*. Saint Lucie Press.
- Österle, H. (1995). *Business Engineering - Prozess- und Systementwicklung*. Springer-Verlag.
- Pohl, K., Böckle, G., and Linden, F. J. v. d. (2005). *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer.
- Reichert, M., Hallerbach, A., and Bauer, T. (2014). Lifecycle Support for Business Process Variants. In Jan vom Brocke and Michael Rosemann, editor, *Handbook on Business Process Management 1*. Springer.
- Rosa, M. L., Dumas, M., ter Hofstede, A. H. M., Mendling, J., and Gottschalk, F. (2008). Beyond control-flow: Extending business process configuration to roles and objects. In Li, Q., Spaccapietra, S., and Yu, E., editors, *27th International Conference on Conceptual Modeling (ER 2008)*, pages 199–215, Barcelona, Spain. Springer.
- Saidani, O. and Nurcan, S. (2007). Towards context aware business process modelling. In *8th Workshop on Business Process Modeling, Development, and Support (BPMDS07)*, CAiSE, volume 7, page 1.
- Sinnhofer, A. D., Pühringer, P., and Kreiner, C. (2015). varbpm - a product line for creating business process model variants. In *Proceedings of the Fifth International Symposium on Business Modeling and Software Design*, pages 184–191.
- Valena, G., Alves, C., Alves, V., and Niu, N. (2013). A Systematic Mapping Study on Business Process Variability. *International Journal of Computer Science & Information Technology (IJCSIT)*.
- Willaert, P., Van Den Bergh, J., Willems, J., and Deschoolmeester, D. (2007). *The Process-Oriented Organisation: A Holistic View - Developing a Framework for Business Process Orientation Maturity*. Springer.
- Zhao, X. and Zou, Y. (2011). A business process-driven approach for generating software modules. *Software: Practice and Experience*, 41(10):1049–1071.

POSTERS

Towards Context-aware Border Security Control

Boris Shishkov^{1,2} and Dimitris Mitrakos³

¹*Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, 1113 Sofia, Bulgaria*

²*IICREST, 53 Ivan Susanin Street, 1618 Sofia, Bulgaria*

³*Department of Electrical Engineering, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece*
b.b.shishkov@iicrest.org, mitrakos@eng.auth.gr

Keywords: Enterprise Modeling, Model-driven Engineering, Context-awareness, Border Security, SDBC.

Abstract: Context-aware systems allow for adapting the system behaviour to the context situation at hand and we have seen good applicability of context-aware systems in domains, such as Mobile Health. Even though this could also be useful for the Border Security domain, applying context-awareness in this domain is not trivial since the possible context situations are numerous and difficult to predict. Still, context-aware Border Security systems are needed as a possible way to overcome the inevitable shortage of resources along the borders. Smooth and fast border crossing for travellers, in combination with adequate level of security, can be achieved if: (i) at any moment, the context situation is properly captured; (ii) there is potential for behaviour and resources (from the Authorities' side) corresponding to each possible context situation. The context situation capturing is about sensors, data streaming, and so on. Establishing the right behaviour / resources is about enterprise modeling and business rules, and it is also about automation that assumes in turn integration of software applications in the overall Border Security system. In this position paper, we address all that, inspired by the SDBC Approach, Enterprise Ontology, Semiotic Norms, and the principles of Context-aware Systems. Reporting research-in-progress, we only present our way of modeling and we identify several domain-specific concerns that are related to the application of the built models. We also provide a list of recommendations that are expected to be useful with regard to possible Border Security system developments.

1 INTRODUCTION

Most software systems need to be adequately integrated in their enterprise environment comprising enterprise modules, human agents, and even already running software applications. Hence, traceability and possible mappings are often needed between a piece of software and an enterprise module and vice versa (Shishkov, 2016); this we call *enterprise-software alignment* (Shishkov and Janssen, 2016). The SDBC Approach allows for realizing such an alignment in a component-based way (Shishkov, 2005), justifying that software specification should be based on corresponding enterprise models. This can be achieved when we have sufficient knowledge of the enterprise environment and this environment is more or less stable. Nevertheless, if we have a constantly changing enterprise environment, we would face a new challenge – the challenge of handling *adaptability*. This means adapting to the surrounding situation in two directions: firstly, if the enterprise environment is changing, then the

enterprise modules “inside” should be changing in turn; secondly, if the enterprise environment is changing, then the “behaviour” of the software applications running inside the enterprise should be adapted as well. Said otherwise, we should be able to adequately establish the situation at hand and provide the right enterprise / software behaviour accordingly. This was called *context-awareness* in research addressing such adaptability-related challenges in the domain of Mobile Health (AWARENESS, 2006). An example is Tele-Monitoring: a patient is being monitored from distance, using mobile technology; there are several possible-to-occur situations, each of them being easy-to-capture distantly; then action is triggered, corresponding to the captured context situation. Thus, we argue that context-awareness is a solution and is easy-to-apply only if we have a limited number of possible-to-occur situations. However, in the current position paper, we address a domain characterized by MANY possible-to-occur situations: this is the Border Security domain. In particular, we are interested in the *monitoring of illegal migration and in combatting related crime*

(and goods smuggling). Guarding along country borders by electronic means, using various channels: infrared images, visible images, proximity sensors, and so on, followed by some kind of intelligent data fusion algorithms, has been addressed in several European Projects, such as the European FP7 Project: “Protection of European Seas and Borders through the Intelligent Use of Surveillance” (PERSEUS, 2015) and the European EBF Project: “Land Border Surveillance – Strengthening of Reaction Capacity” (LandBorderSurveillance, 2012). Still, those efforts appear to be insufficient as it can be seen from the severe border problems in Greece and Italy, in 2015-2016 (FRONTEX, 2016). We recognize the need for better interoperability with regard to the existing national border security platforms and systems. We do agree that innovative capabilities are needed, including trans-national exchange of available and useful information, as it has been discussed in the above-mentioned projects. Nevertheless, we argue that those would only be part of the solution related to the challenge of improving Border Security, supported by (software) technology. We claim that context-awareness is not applied of full value at the borders and we claim that context-aware Border Security systems would make a difference in the direction of improvement. At the same time, we go back to our previous conclusion, already mentioned, that it is not straightforward applying context-aware solutions in the Border Security domain. Hence, research is needed on *Context-Aware Border Security* (CABS) control, as a possible way to overcome the inevitable shortage of resources along the borders (it would be difficult for a country to supply persons and equipment at every potentially risky border point). A CABS system would guarantee adaptability with regard to the situation at hand – persons and equipment would only be supplied at the spot where they are needed and in the moment when they are needed. Of course, if total “tension” appears at all risky border points at the same time, such an approach would “crash” but this is similar to the situation of all customers of a bank claiming back their deposits at the same time. Such situations are considered to be of low probability to occur and are thus left beyond the scope of this paper.

Hence, we claim that smooth and fast border crossing for travellers in combination with adequate level of security can only be achieved if:

- (i) at any moment, the context situation is properly captured;
- (ii) there is potential for behaviour and resources (from the Authorities’ side) corresponding to each possible context situation.

The context situation capturing is about sensors, data streaming, and so on. Establishing the right behaviour / resources is about enterprise modeling and business rules, and it is also about automation that assumes in turn integration of software applications in the overall Border Security system.

Further, because of the increased complexity with regard to Border Security situations and occurrence probabilities, we need to address *data aspects* (going beyond just interoperability and information exchange, see above). Ways to capture data, quality of data and the probability that the captured data is correct, reliability, versioning, privacy, and so on, are of importance as well. Hence *data analytics is to be integrated in the enterprise modeling and also in the software development* in order to facilitate context-awareness, especially in the Border Security domain.

In this position paper, we address all this, inspired by the SDBC Approach (Shishkov, 2005), Enterprise Ontology (Dietz, 2006), Semiotic Norms (Liu, 2000; Shishkov et al., 2006), and the principles of Context-aware Systems (AWARENESS, 2006).

Reporting research-in-progress, we only present our way of modeling (Section 2) and we identify several domain-specific concerns that are related to the application of the built models (Section 3). We also provide (as part of the conclusions) a list of recommendations that are expected to be useful with regard to possible Border Security system developments.

2 WAY OF MODELING

As mentioned above, we address the challenges of deriving software and integrating it in its enterprise environment, inspired by the SDBC Approach (Shishkov, 2005). This in turn assumes reference to the theories of *LAP – Enterprise Ontology* (Dietz, 2006) and *Organizational Semiotics* (Liu, 2000); those are briefly outlined in another paper published in the current proceedings (Shishkov & Janssen, 2016). The idea behind SDBC is that (re-usable) enterprise modeling constructs (called “Business CoMponents”) are identified and reflected in corresponding software specification models to be in turn decomposed in terms of models of software components. SDBC is consistent with the principles of *Model-Driven Engineering – MDE* (Schmidt, 2006): building a technology-independent model goes first, then it is to be decided what would be automated, and in the end is the software derivation. This is the “basis”, no matter if we go for developing a context-aware system or a system that is not

context-aware – in order to contribute especially in the direction of context-awareness and particularly in the Border Security domain, we need to have a “valid” basis to start to build upon. For that we take SDBC not only because developing this approach is part of our previous work but also because SDBC has been validated by means of case studies carried out in the domains of Finance and Healthcare.

Further, the SDBC Approach and the principles of MDE were successfully applied in specifying context-aware systems in domains, such as Mobile Health.

Nevertheless, as mentioned already, the Border Security domain assumes greater complexity because of numerous possible situations and prediction difficulties. Further, what is observed at the border is a “mixture” of personnel and devices, subject to numerous rules and functionalities (FRONTEX, 2016), and it is not trivial approaching this in terms of technology-independent models, automation, and so on. This is because some (intuitive) tasks can only be realized by humans while other (surveillance) tasks can only be realized by devices, to give just an example. Hence, we need to “adapt” SDBC to the peculiarities of the Border Security domain. SDBC goes “top-down”, from a “bird-view” enterprise model through delimitation with regard to the software system –to-be, to implementation. Nevertheless, for specifying a CABS system, we propose to go “middle-out”, as exhibited on Figure 1, and we adapt the application of SDBC accordingly.

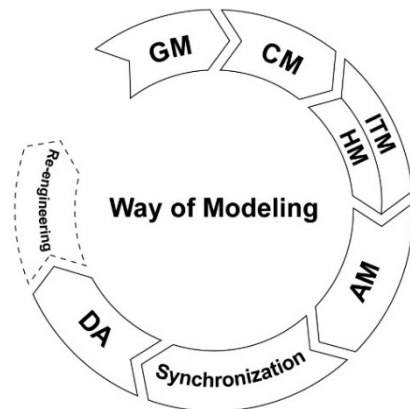


Figure 1: CABS – Way of Modeling.

On the Figure: “GM” stands for “*general model*”, “CM” stands for “*conceptual model*”, “ITM” stands for “*IT model*”, “HM” stands for “*humans model*”, “AM” stands for “*aspect model*”, and “DA” stands for “*data analytics*”.

We propose to go “middle-out” because in the Border Security domain, it seems most pragmatic to

start with modeling “what is there” (a mixture of person-tasks, device functionalities, and so on to be seen at the border) – such a model we call a *general model* (GM). No other model that would inevitably be abstract, would allow for grasping everything correctly and also communicating it adequately with all relevant stakeholders – this is claimed to be of great importance particularly for the Border Security domain. Just as an example of GM, we consider a typical point at an external EU border, the border between Bulgaria and Turkey (FRONTEX, 2016), and we take an “imaginary” view on things that may be seen at a border point – see Figure 2.

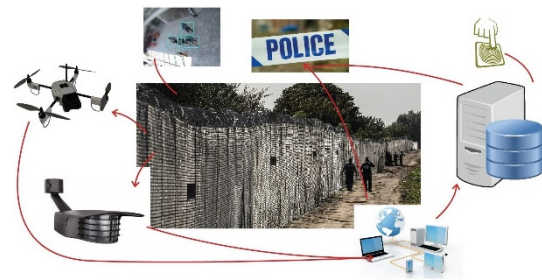


Figure 2: GM Example.

As seen from the Figure: there is a border fence and border police officers patrolling along the fence; there are cameras attached to the fence, which realize crowd monitoring and there are mobile cameras attached to drones; there are finger-print devices that can be used by police officers for personal identification; there are (networked) computers running and streaming all sensor raw data, and also processing it by applying data fusion algorithms (for example), allowing “higher-level” reasoning, and so on. Hence, we claim that such a model should be the starting point in specifying a CABS system.

We use the GM as basis for deriving a CABS-related classification of concepts – this we call a *conceptual model* (CM) – see Figure 1. This way of “arriving” at the CM guarantees that our further system development activities would be “grounded”. The human agent concept and the device concept appear to be essential within the CABS conceptual model (Figure 3). That is because the CABS general model suggests that anything that can be observed at the border either relates to a personal (human) role or to a functionality delivered by a device (equipment). Further, among the human agents at the border (besides the persons who are crossing the border and are thus left outside the scope of the CABS system) are customs officers and police officers, while among the devices one could observe at the border are sensors,

computers, and vehicles. Sensors in turn could be audio sensors and video sensors, while computers could be servers and personal computers, and vehicles could be cars and drones. And so on. This is just as an example on how a CM can be derived, based on a GM.

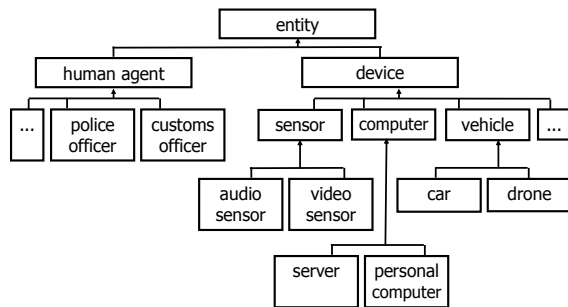


Figure 3: Deriving a Conceptual Model.

Such a conceptual model is the necessary starting point in an SDBC software development but also if one would just need to build an enterprise model.

What goes next, as it can be seen from Figure 1, is splitting in two, namely: *humans model* and *IT model*. Said otherwise, all that concerns persons at the border is put on one side and all that concerns devices at the border is put on another side. We do not mix this up even though in enterprise modeling we often mix up activities that are realized by humans and activities that are automated, realizing that it is possible to further automate activities currently realized by humans. For example, the SDBC Approach was used to specify the automation of what a human actor (in particular: Insurance Broker) was doing, for the sake of “replacing” humans by software (Shishkov, 2005). In such cases, it is straightforward putting together in the model issues that concern human actors and issues that concern non-human actors. This is claimed to be nevertheless inappropriate with regard to the Border Security domain and that is why the CABS conceptual model strongly distinguishes between human agents and devices (see Figure 3). Hence, as according to Fig. 1, we model the human entities and processes (HM), and in parallel, we model the devices-related entities and processes (ITM).

That is because devices at the border (on one hand) capture raw data, stream data, run algorithms, and so on, and all those issues concern the electronics features of the devices, not even so much the software in some cases; further, those societally elementary but computation-intensive tasks assume nothing like real-life communication, intuitive behaviour, pro-activity, and so on. Persons at the border (on the other hand),

such as border police officers, for example, are often valued especially for their intuitive behaviour, enriched by many years of experience. It is the person (not the device) who can “smell in the air” anything wrong and possibly trigger a check without even being able to explain why. This is in contrast to other domains where sometimes a task can equally be completed by a human actor or by a non-human actor. We claim that at the border, what persons do and what devices do are “two different Worlds”.

What is next (see Figure 1) is the modeling elaboration – each of the two models (HM and ITM) is to be elaborated; having provided just a classification of concepts is not enough - statics (entities and their relationships), dynamics (processes and states), and so on, need also to be provided as elaboration. Such elaboration models are called *aspect models* (AM). We build those models, inspired by the four modeling perspectives of the SDBC Approach: *Structural Perspective* that reflects entities and their relationships (Dietz, 2006); *Dynamic Perspective* that reflects the overall business process and the states of each entity (evolving accordingly) (Van der Aalst, 2011); *Data Perspective* that reflects the information flows across entities within the system and flows reaching beyond the system’s boundary (Shishkov, 2005); *Language-Action Perspective* that reflects real-life human communication and expression of promises, commitments, and so on (Shishkov et al., 2006).

Hence, as elaboration, we build structural models, dynamic models, data models, and language-action models – those are four aspect models. We proceed as follows:

- With regard to the HM we do all four elaborations, but
- With regard to the ITM, we do all except the last one – the language-action model, because devices cannot express commitments, do promises, and so on, using language.

Then, after having elaborated the HM and the ITM in terms of aspect models, we need to “synchronize” between the two, as depicted in Figure 1. By bringing together the two structural models, we establish all relationships between human entities and device entities, by bringing together the two dynamic models, we establish if any human action requires as pre-condition a corresponding device action to be completed (and vice versa), and so on. A question could be asked why was it necessary to firstly split and then synchronize. By firstly splitting in two (persons and devices) and then bringing them together, we guarantee for:

- Adequate modeling, because each model is being generated based on the right background (when modeling humans, we consider humans and when modeling devices, we consider devices);
- Exhaustive modeling, because in the end the “two parts” are brought together.

Then comes the *data analytics* (DA) – see Fig. 1 and it is important not to confuse this with the “data perspective” of SDBC, reflected in deriving the aspect models. This perspective has a purely functional drive – we consider the data flows as part of the delivered functionality (for example, a balance needs to be checked, before allowing a person to withdraw money from his/her account). The DA, on the contrary, has a non-functional drive and is crosscutting. For example: (i) Extracting a face image of a person, allows for realizing activities on that basis but is it legally correct to do so and if not, is this information an “input” or not? (ii) If what we get as sensor raw data has, say 80% trustworthiness, what we do forward? Those are just two examples and we would also mention data versioning, obsolete data, wrong approximations, and so on. Nevertheless, just staying aware of those issues is not helpful – we need to address them functionally (even though in essence they may be non-functional). A possible way to do this is by applying Semiotic Norms – see another paper in the current proceedings (Shishkov & Janssen, 2016), for example (on considering crowd monitoring):

Whenever the System has established a situation
if the level of trust is less than 75%
then the System
is obliged to check also another source.

And in the end comes *re-engineering* – Figure 1, and the line there is dashed, to indicate that it is not always needed to re-engineer the system. Still, if the built models help in identifying inconsistencies, re-engineering could follow straightforwardly.

This is our proposed way of modeling CABS systems, in line with the SDBC Approach and as already mentioned, in the next section, we reason about the application of the CABS models, by identifying several concerns related to such an application.

3 DOMAIN-SPECIFIC CONCERNS

Being successful in the modeling phase is crucially important, no question about this, but it is equally

important to be successful in reflecting the models in architectures that are an adequate basis for implementation. In this regard, we have identified several *domain-specific concerns*, not claiming exhaustiveness nevertheless, which are presented in the current section; those concerns are *intuitive behaviour*, *devices’ technical restrictions*, *security*, and *privacy*.

3.1 Intuitive Behaviour

As mentioned before, a border police officer is sometimes especially valued for his/her capability to be intuitive with regard to a situation, applying a “sixth feeling” in deciding what to do. In our view, this is nearly impossible to capture and reflect in models. This means in turn that an important Border-Security-related “asset” would inevitably remain outside the “scope” of a CABS system.

3.2 Devices’ Technical Restrictions

Often a device at the border is a piece of hardware and its electronics would often be restrictive with regard to the ways in which it could be used. If a software application would have to “bridge” the device to the system, then in-depth knowledge on the electronics of the device would be a must – this complicates the job of enterprise modelers and software designers. Those restrictions are nevertheless not only electronics-related – a drone, for example cannot “carry” more than half a kilo and cannot stay in the air more than a certain amount of time – such restrictions should be taken into account as well.

3.3 Security

A CABS system should follow highest security standards, “higher” than even in Banking because a Border Security failure may lead to dramatic consequences for hundreds and thousands of people. This assumes not only establishing advanced computer networking but also “guaranteeing” what human actors would do (or not do). This is to be taken into account and in our view, it is very difficult to actually establish and maintain so high security standards in two perspectives - personal and technical.

3.4 Privacy

A CABS system should be a privacy-sensitive system because of a number of privacy-related risks at the border, concerning both border police officers and

persons crossing the border. For example, it should not be possible that terrorists whose crossing the border was obstructed by a border police officer, are able to later on identify the police officer. Another example: it should not be possible that crowd monitoring information is used later on in another context, with regard to the monitored person(s). Those issues are “burdened” with many legal aspects and what makes things even more complicated is that legislation differs from country to country, even inside the EU.

Those are four important concerns, related to the application of the CABS models and they indicate that there are issues beyond enterprise modeling and software design, that need to be taken into account and have great potential impact on the CABS system –to-be.

4 CONCLUSIONS

There is room for improving both the effectiveness and efficiency of the control at the external EU borders, and context awareness is a desired feature in this regard. Nevertheless, the development of context-aware Border Security systems is not trivial because of numerous possible-to-occur situations and prediction difficulties, and because of person-specific (intuitive) and device-specific (algorithmic) behaviour patterns. A “middle-out” realization of the SDBC Approach is proposed to tackle this, coming through the derivation of a general model, a conceptual model (split in two – a humans model (HM) and an IT model (ITM)), modeling elaborations, a synchronization between HM and ITM, and enrichment in terms of data analytics. Persons’ intuitive behaviour, the technical restrictions of devices, security, and privacy are among the concerns related to the application of those models. Inspired by the aim of furthering this research, we would stick to the following recommendations that we identified accordingly:

- A CABS system should be modeled as a human-centric system because the intuitive behaviour of border police officers is and will be of great importance at the border.
- In using devices, the quality-of-data is to be of great importance for a CABS system, and this issue is to be also handled functionally.
- A CABS system is to be modeled as a context-aware system where the delivered system behaviour depends on the context situation at hand.

- A CABS system is to be based on an enterprise model that is split in two – one part featuring persons and another part featuring devices; a synchronization between the two is essential.
- Security and privacy are issues that are to be taken into account additionally, in order to have a CABS system of real value.

REFERENCES

- AWARENESS, 2006, the website on the Freeband AWARENESS Project: https://www.utwente.nl/ctit/research/research_projects/concluded/bsik/freeband/projects/awareness
- Dietz, J.L.G., 2006. *Enterprise Ontology, Theory and Methodology*. Springer-Verlag, Berlin Heidelberg.
- FRONTEX, 2016, the website on the European Agency FRONTEX: <http://frontex.europa.eu>.
- LandBorderSurveillance, 2012, the website on the EBF LandBorderSurveillance Project: http://ec.europa.eu/dgs/homeaffairs/financing/fundings/projects/project_example_048_en.htm
- Liu, K., 2000. *Semiotics in Information Systems Engineering*. Cambridge University Press, Cambridge.
- PERSEUS, 2015, the website on the European FP7 PERSEUS Project: http://cordis.europa.eu/project/rcn/97515_en.html
- Schmidt, D.C., 2006. *Model-Driven Engineering*. IEEE Computer Society.
- Shishkov, B., 2016. Foreword. In *BMSD'16, 6th International Symposium on Business Modeling and Software Design*. SCITEPRESS.
- Shishkov, B., 2005. *Software Specification Based on Reusable Business Components (PhD Thesis)*, TU Delft – SIKS Publishing. Delft.
- Shishkov, B., Dietz, J.L.G., Liu, K., 2006. Bridging the Language-Action Perspective and Organizational Semiotics in SDBC. In *ICEIS'06, 8th International Conference on Enterprise Information Systems*. SCITEPRESS.
- Shishkov, B. and Janssen, M., 2016. Towards a Service-Oriented Architecture for eVoting. In *BMSD'16, 6th International Symposium on Business Modeling and Software Design*. SCITEPRESS.
- Van der Aalst, W., 2011. *Process Mining - Discovery, Conformance and Enhancement of Business Processes*. Springer-Verlag, Berlin Heidelberg.

Introduction of BEMES, a Webtool to Simplify Business Process and Equipment Cost Modelling

Jonathan Spruytte, Marlies Van der Wee, Sofie Verbrugge and Didier Colle
*Department of Information Technology, Ghent University – iMinds, iGent-Toren,
Technologiepark-Zwijnaarde 15, B-9052 Gent, Belgium
{jonathan.spruytte, marlies.vanderwee, sofie.verbrugge, didier.colle}@intec.ugent.be*

Keywords: Techno-economic, Equipment Cost Modelling, ECMN, Process Cost Modelling, Business Process, BPMN.

Abstract: Modelling both business processes and equipment costs is an elementary step in the techno-economic evaluation of a business case (infrastructure project, network deployment, etc.). Most of a upfront business case evaluation, on which strategic decisions are based, relies on inflexible and error-prone spreadsheet models. In order to provide a stable and reliable alternative, this paper presents the equipment coupling modelling notation (ECMN). ECMN allows calculating the costs for equipment installation using dynamic, easy-to-understand, graphical equipment trees. For business processes, we developed an adapted version of the business process modelling notation (BPMN). In order to simplify the usage of both ECMN and BPMN, we also present the BEMES-tool (Business Modelling and Simulation). BEMES is a graphical web-based tool that can be used to draw both models using drag-and-drop and calculate the models resulting in visual output. The BEMES-tool consists of a set of interlinked modules; we describe the functionality of each of these modules and discuss the benefits this strong modularized approach yields for both the current and future features.

1 INTRODUCTION

The techno-economic evaluation of a network deployment planning problem can be split in 4 major components: scope, model, evaluate and refine as discussed extensively in (Verbrugge, 2009). In the modelling step, the input collected during the scope phase is used to model both costs and revenues. These models serve as input for the evaluate phase, which combines the outcome of both business process and equipment cost models with revenue estimates to evaluate the business case for one or multiple stakeholders. The fourth and final phase, refine, applies game theory, real options and sensitivity analysis to capture the impact of uncertainty. The techno-economic research group at Ghent University (www.technoeconomics.ugent.be) has developed several Java-based tools to automate (parts of) these modelling phases (Tahon, 2014), (Kasier, 2009).

Within the context of this paper, two modelling languages which simplify cost modelling will be discussed; the distinction is made between models for equipment and models for processes.

For process modelling, the well-known Business Process Modelling Notation (BPMN) can be used

(White, 2004). BPMN uses a flowchart-like notation to model the sequence of actions and decisions in a business process; an example is provided in Fig. 1.

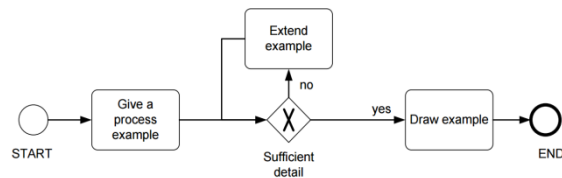


Figure 1: A basic process with 1 decision and 3 actions modelled in BPMN.

A lot of planning projects require reliable estimations of the Bill of Material (BOM) for all equipment that needs to be installed (e.g. installation of a central office, dimensioning of a data center, etc.). The BOM lists all equipment and offers a view of the total equipment cost of a planning project. There however is no formal, standardized approach to equipment modelling (to draft a BOM) so far. A typical equipment cost model approach starts from a hierarchical equipment tree, interlinking equipment (e.g. each server needs a slot in a server rack). This approach is often based upon a large amount of sheets

in your favourite spreadsheet software package and may or may not include additional tools (Oase, 2011). This approach is not only time-consuming, it also offers little flexibility and makes calculating the same model for a variety of input a tedious job. For tackling these issues, we have developed the Equipment Coupling Modelling Notation (ECMN) (Casier, 2014).

Using ECMN, it is possible to graphically draw the equipment tree, add granularities to each branch in the tree and link cost-drivers to each piece of equipment (Van der Wee, 2012). A cost-driver is a time-dependent function, which serves as an input for the model and can be linked to any element of a model.

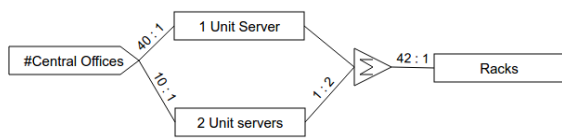


Figure 2: An example ECMN-model consisting of 1 cost driver and 3 pieces of equipment.

Figure 2 shows a basic ECMN-model which consists of a total of 5 elements, from left to right:

- *#Central Offices*: is a driver and is the only input considered.
- *1 Unit Server* and *2 Unit Servers* are as the name indicates servers that respectively require 1 and 2 units in a rack. As the links (granularities) indicate, one central office requires 40 1-Unit Servers and 10 2-Unit Servers.
- A *summation*, which sums the total of required rack-space of both types of servers. The granularity of the links is respectively 1:1 and 1:2 as the 2-Unit Servers requires two spaces in a rack.
- *Racks* is the last piece of equipment; one full-sized rack typically has 42 units, so the granularity of the link has been set accordingly. (Note: the resulting amount of a link is always round to the next integer, if the start of the link is 5 with a granularity of 3, the end of the link will be 2. This fits within the equipment installation reasoning: as soon as you have an extra server that no longer fits in the first rack, you have to install a second rack).

When giving a value to the driver (*#Central Offices*), the quantity of both types of servers and number of racks is automatically calculated with the corresponding cost. A number of additional parameters (e.g. reinstallation period) can be provided which will influence the total cost.

In order to easily create both ECMN and BPMN models, we have created the BEMES-tool which is further discussed in this publication. The remainder of this publication is structured as following: in section 2 we discuss the modular approach of the tool and the benefits it yields. In section 3 we introduce a public test version of the tool, which allows everyone to try out the BEMES-tool. Finally, section 4 briefly summarizes this paper and presents further steps for the BEMES-tool.

2 THE BEMES-TOOL

In order to incorporate both ECMN and BPMN in the modelling phase we developed a graphical web-tool: BEMES (Business Modelling and Simulation). The BEMES-tool allows us to create both ECMN and BPMN models using simple drag-and-drop-actions and consists of three interlinked modules (Figure 3): (1) the graphical web-based frontend (the editor, which offers separate views for ECMN and BPMN), (2) the repository that stores the models and (3) the calculator hub linked to a set of calculators that perform the actual calculations of the models and return the result.

Between these modules, the models are exchanged in XML-format using the REST-protocol.

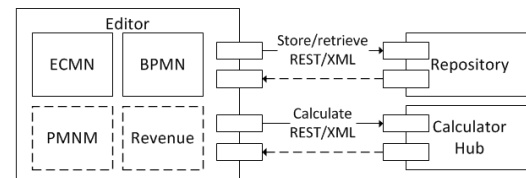


Figure 3: The BEMES-tool has a modular approach. Interaction between the modules uses the REST-protocol and the XML-format.

As an addition to these modules, a Java-interface has been created. This interface can retrieve, extend/modify and calculate models via the Java programming language. In the next paragraphs, the functionality of the modules is summarized and the modular approach of the BEMES-tool is discussed.

2.1 Editor

The first module is the editor which provides the graphical web user interface and interlinks behind the scenes with the Repository (2.2) and the Calculator Hub (2.3).

After authentication, a user has the ability to switch between a number of views, depending on the

wanted type of model (ECMN or BPMN). Using the editor, a user can draw an entire model using drag and drop, change parameters of objects, calculate a model and get visual output without having to write one single formula or write any code.

2.2 Repository

The repository is responsible for the storage of - and the access to - the different models. The current version of the repository stores the models directly, in XML-format, on disk. Since at the moment only one repository is used and models are directly linked to a single user, using a file-based storage entails no problems. In the future, once models may be shared between users and concurrent edits may be possible, any kind of database should be used to simplify the synchronisation process.

2.3 Calculator Hub & Calculators

The Calculator hub is the single point of entry to calculate models. The calculator hub receives a request to calculate a model (represented in XML). Based upon the type of model, it selects a matching calculator and forwards the request, as shown in Figure 4.

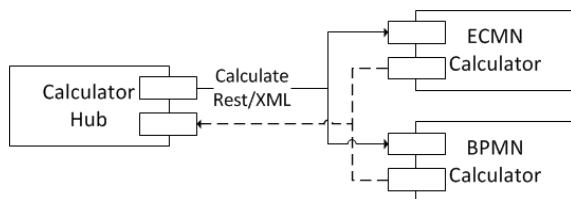


Figure 4: The Calculator Hub links to a set of calculators.

This single point of entry approach has a number of benefits.

- **Single Point of Control:** since any model passes through the Calculator hub, it is the perfect location to validate the incoming XML-stream.
- **Flexible and Extendable:** based upon the incoming XML-stream, the Calculator hub selects the correct calculator. At the moment, only one calculator per type is deployed, so this forwarding is straightforward. In the future, multiple calculators per type may be deployed and the Calculator hub may be used for load balancing.
- **Simplified Backward Compatibility:** instead of having calculators that support every version of the XML-format, they should only support one (major) version. This makes the implementation of each calculator less boated and thus much

simpler. When changing to a new major version of the XML-format, the old calculators which are based upon the older format can simply stay online. When a request is made to the Calculator hub, it will select a matching calculator based upon the type and (major) version of the model.

- **Access Control:** the Calculator hub is the perfect location to implement any rate limits (e.g. number of model calculations per hour) or may in the future be used to limit access to the calculation of some model types (e.g. licensed access to modules).

2.4 Java-to-BEMES-Interface

The last module, the Java-to-BEMES-interface allows access to models designed in the editor via the Java programming language. Using the Java-to-BEMES-interface, a user can load a model, override model values (e.g. the cost of an equipment item) and calculate the modified model. To do so the, the interface links directly to both the repository and the calculator hub, in a similar fashion as the Editor, as shown in Figure 5.

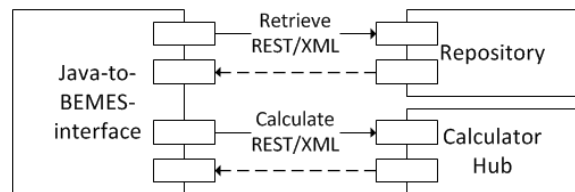


Figure 5: The Java-to-BEMES-interface allows the retrieval, modification and calculation of a model drawn in the Editor.

While the graphical user interface is great for building the model and receiving basic output, the Java-interface is great for running a model with a variety of input values e.g. sensitivity analysis or scenario analysis. Note that, currently, this Java-interface is limited to ECMN-models.

2.5 Benefits of the Modular Approach

As said, the BEMES-tool consists of multiple modules; this clear division yields a number of merits:

- Modules can easily be distributed across multiple servers; this way, modules can easily be duplicated and accessed using any load balancing scheme (e.g. round-robin).
- Modules can easily be re-implemented, as long as the existing REST-interface is respected (e.g. a

new repository which stores models in any kind of database).

- Any module can be considered as a standalone-module for development. As long as the XML-format is respected, any module can be changed/extended without any impact for the other modules.
- Any additional modules can be linked via the calculator-hub with minimal effort.

3 TEST CASE: LIVE DEMO

To show the possibilities of the BEMES-tool, a live demo has been made available at <http://bemes.atlantis.ugent.be/EditorDemo>. A guest-account has been created, to log in use guest for both the username and password.

When logging in, the demo-model as discussed in paragraph 1, should automatically be loaded, if not the model can be loaded using the navigation bar on top. Using this demo the functionality of the BEMES-tool can be tested; it allows the creation of both ECMN and BPMN models. In this demo-version, saving a model to the repository has been disabled.

4 SUMMARY AND FUTURE WORK

Modelling equipment and process costs is an important part of a project business case evaluation (e.g. network deployment, data center dimensioning, etc.). BPMN is the standard language for process modelling, but equipment models are often made using spreadsheets which are error-prone and rather inflexible to calculate for a wide variety of values. In order to simplify this process, we have developed the Equipment Coupling Modelling Notation (ECMN) which is a graphical notation which allows the modelling of an equipment tree. In order to incorporate ECMN in the modelling process we have created BEMES. BEMES (Business Modelling and Simulation) is a graphic webtool specifically developed to draw both ECMN and BPMN models using very simple drag and drop-actions.

Internally BEMES has a strong modular approach resulting in added flexibility and extendibility. Future work includes further development of the Java-to-BEMES interface for BPMN, and inclusion of automatic sensitivity analysis for both type of models.

Apart from BPMN and ECMN, the techno-economic group is also developing models for

network infrastructure deployment cost (PNMN) as well as revenue modelling based on different types of pricing strategies. The goal is to incorporate these models in the BEMES-tool, so the tool can be used to model each part of a (network deployment) business case evaluation.

REFERENCES

- Verbrugge, S. et al., 2009. White paper: Practical steps in techno-economic evaluation of network deployment planning.
- Tahon, M. et al., 2014. Real options in telecom infrastructure projects - A tutorial. *IEEE Communications Surveys and Tutorials*, 16(2), pp.1157–1173.
- Casier, K., 2009. Techno-Economic Evaluation of a Next Generation Access Network Deployment in a Competitive Setting. Ph.D thesis, Ghent University.
- White, S. a., 2004. Introduction to BPMN. *BPTrends*, pp.1–11.
- Van der Wee, M. et al., 2012. A modular and hierarchically structured techno-economic model for FTTH deployments. 16th international conference on Optical Networking Design and Modeling, *Proceedings*, (1), pp.1–6.
- OASE, 2011. Deliverable 5.1 – Overview of tools and methods. <http://www.ict-oase.eu/index.php?page=120&>

Using a Multi-Perspectives Approach for Building a Socio-Technical Information System

Sayeb Belhassen Yemna, Foughali Ibtissem and Hajjami Ben Ghezala Henda
RIADI Lab., The National School of Comp. Sciences, University Campus of Manouba, Tunisia
{yemna.sayeb, alfoughaliibtissem}@gmail.com

Keywords: Socio-technical Information System, BPM, BPMO.

Abstract: In this paper we propose an approach for developing Socio-Technical Information Systems - STIS with integrating social, business, and technical needs of an enterprise. As a first aim of our proposed approach is to ensure that both technical and human factors are given equal weight in the design process. As a second objective is to provide the right information at the right place and the right time and thereby enable communication between people, machine and computer, and their efficient cooperation and coordination. To build an STIS, we propose a multi-perspective approach based on 4 perspectives, namely: usage (resources) perspective, ESB perspective, business process modeling perspective, and SOA perspective. We use the Business Process Modeling Ontology in considering the different perspectives.

1 INTRODUCTION

When developing an information system, we need adequate underlying enterprise models that would help appropriately considering the real-life business context in which the information system would have to operate. This challenge is being addressed by numerous researchers whose efforts have been inspired by the goal of closing the gap between enterprise modeling and software design. Enterprises have to become agile, sensitive to changes in market forces, and capable of responding with incremental modifications in business and services provided (adaptation) as well anticipating radical changes by responding with new and breakthrough business models (innovation).

Recent trends in the development of information systems are associated with intention to consider the social system and the technical system with equal weight.

On the one hand, the need of integrating enterprise systems and technical systems calls for novel ICT framework models and solutions, such as “*EIS*” - *Enterprise Information Systems* which refer to organizational implementations of commercial software packages that enable the integration of transaction-oriented data and business processes through an organization (Markus and Tanis, 1999). But to date there is a wrong understanding of those systems. The prominent current idea is that an

enterprise information system is a product like for example a car (Dietz, 2014). We aim at re-established people who work on the enterprise as the ‘parts’ of every organization. They deliver services to each other and to the environmental actors through transactions. Those transactions are generally represented through business models which describe the rationale of how an organization creates, delivers, and captures value.

Our main objective in this paper is to propose a Socio-Technical Information System approach – STIS, which integrates the social, business, and technical needs of an enterprise. As a first aim of our proposed STIS is to ensure that both technical and human factors should whenever possible be given equal weight in the design process. As a second objective is to provide the right information at the right place and the right time, and thereby enable communication between people, machine and computer and their efficient cooperation and coordination. (Mumford, 2006). A socio-technical system differs from other types of systems in that it has human participants performing essential tasks inside the system. However, as any other system, a socio-technical system can be modeled as consisting of a number of components interacting with each other, and the system’s environment.

A socio-technical system is considered as composed of two sub-systems: a technical system and social system. The technical sub-system encompasses

both technology and processes. The social sub-system encompasses the people who are directly involved in the information system's structure in which they are embedded (Rabah Imache, 2012).

Our STIS approach is in line with the Service-Oriented Architecture – SOA (Rafati, 2013).

To build an STIS, we propose a multi-perspective approach based on a usage (resources) perspective, a business process modeling perspective, a SOA perspective, and an integration perspective. We use the Business Process Modeling Ontology – BPMO, in considering the different perspectives. In Section 2 of this article, we will present the proposed approach and we will give more details about the four mentioned perspectives, and also about the role of BPMO. In Section 3, we will illustrate the technical architecture for building an STIS, based on SOA and Enterprise-Service-Bus technologies. And we conclude the article with the conclusion.

2 PROPOSED APPROACH

We start with the business process modeling perspective which is considered essential. As presented by (Chen, 2012) a business process consists of workflows which are a series of interested tasks with data flowing and processing. It can be a person-to-person, person-to-system, system-to-system or a combination of three. Business modeling is not only about the business processes but also about the construction that concerns the business strategy and the design of organizational structures. The strategy formulation is based on resources and capabilities. The Object Management Group – OMG propose the Business Process Modeling Notation - the BPMN 2.0 standard to support business process modeling (Chinosi and Trombetta, 2012). Although the syntax of BPMN is clearly defined, semantics is not (Guedria, 2014). Since 2008, the OMG have tried to solve this problem, by the defining “semantics of business vocabulary and business rules” (SRVR) (OMG 2008).

We argue that business process management requires a comprehensive integration among a variety of systems and services. Many authors have noticed the importance of business aspects of Enterprise Integration (Li and Williams, 2004) and have tried to use it when building a business process model.

Vernadat (1996) gave his definition as follows: “Enterprise integration is concerned with facilitating information, control and material flows across organization boundaries by connecting all the necessary functions and heterogeneous functional

entities (information system, devices, applications and people) in order to improve communication, cooperation and coordination with this enterprise so that the enterprise behaves as an integrated whole, therefore enhancing its overall productivity, flexibility and capacity for management of change”.

In 2003, the needs for physical connectivity in the area of enterprises have grown into a much bigger and richer perspective, Enterprise Interoperability (EI), which is about both information and functionality sharing between concerned parties (Vernadat, 2003).

We consider SOA as an integrated enterprise model which can be applied to transform an enterprise model to a flexible one in order to respond to several important trends such as globalization (Rafati, 2013).

We hope also to consider the role, competences and needs of the human resources who are concerned with the integration model. That's why we consider a usage perspective called also use resources perspective. We consider this perspective as important in building an STSI. We illustrate in Figure 1, the concept of actor which is defined as an organizational unit with expressible and collective knowledge, and have competences that reflect the implementation of this knowledge in an operational context, and assigned to a role within a business process (Jabloun Marwen; Sayeb Yemna, 2013).

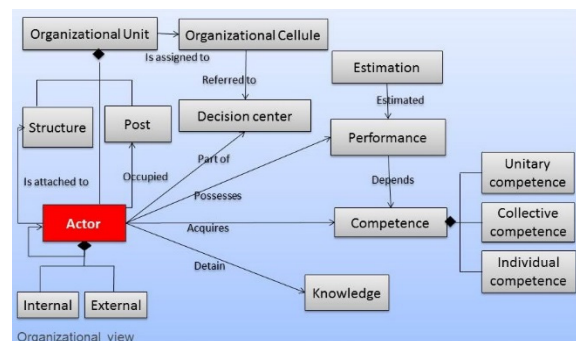


Figure 1: The use resources perspective.

So, to take into account all those perspectives: web services integration and use resources, we use BPMO. The business process modeling ontology is part of an approach to modeling business processes at the semantic level (Lalehrafati, 2013), integrating knowledge about the organizational context, workflow activates, and semantics.

BPMO describes a rich business process model, as demanded by the Business Process Management Community, using ontological descriptions to capture workflow and organisational concerns in a uniform and extensible manner, and reuses the results of

Semantic-Web-Services –related research for the description of interaction activities. There are various advantages for using BPMO. As explained in (Cabral and Liliana, 2009): Firstly, BPMO provides comprehensive semantic annotations for business processes that can be used for automated inference at the business level while facilitating the translation to the execution level. Secondly, BPMO provides links from the process to organisational aspects, which can be modeled independently for different domains. Thirdly, BPMO can be used to verify at the semantic-level restrictions applied to the workflow or certain process activities. Finally, BPMO facilitates the modeling of new (or mediation) processes, based on existing ones as well as the discovery of services for goal-based activities.

The four perspectives for building an STIS are depicted in Figure 2.

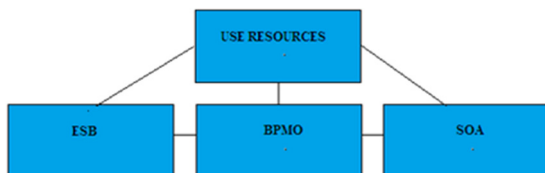


Figure 2: The four perspectives for building an STIS.

3 IMPLEMENTATION

SOA, considered with regard to the implementation, proposes an integrated model based on a central registry which is called broker (Chen, 2012). SOA makes the integration process easier as it would not require software re-development.

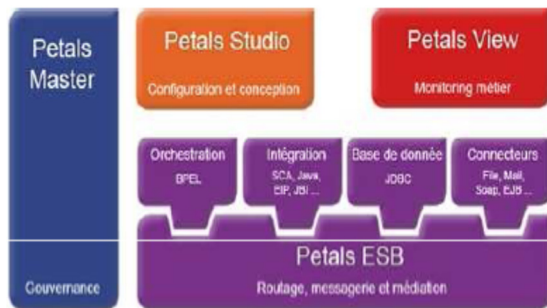


Figure 3: Petals Environment.

Enterprise Service Bus - ESB is a bus architectural technology to connect many disparate systems and services together. It is organised on set of service containers inter-connected over message-based bus. The ESB technology, based on SOA, improves the central registry mechanism of SOA to service

containers architectures. So we decide to experiment the Petals SOA which is a set of tools providing a complete range of software to create, manage and monitor enterprise SOA solutions. *Petals* propose a context and role management through its containers. As illustrated in Figure 3, all tools adhere to open standards and proven use, robust, high performance technologies.

4 CONCLUSION

In this paper we have proposed a multi-perspective approach for building a socio-technical information system (STIS). Our objective was to use the Business Process Modeling Ontology (BPMO) in considering the different perspectives. BPMO describes a rich business process model using ontological descriptions to capture workflow and organisational concerns in a uniform and extensible manner. To implement our approach, we use the Petal ESB environment.

REFERENCES

Cabral, Liliana; Norton, Barry and Domingue, John (2009). *The business process modelling ontology. In: 4th International Workshop on Semantic Business Process Management (SBPM 2009)*, Workshop at ESWC 2009, 1 June 2009, Crete, Greece.

Chinosi, M. and Trombetta, A. (2012). *BPMN: An introduction to the standard. Computer Standards & Interfaces* 34(1):124-134, 34(1):124-134.

Jan L. G. Dietz. *Why ERP Systems Will Keep Failing. ICEIS* (1) 2014: IS-9.

Jabloun M, Sayeb Y, *From EA to Actor Model: knowledge in the service of IS reform Germany*, WISSORG 2013, Mars 2013.

Guédria W. (2014). *Towards an Integrated Model for Enterprise Interoperability. In Proceedings of the Fourth International Symposium on Business Modeling and Software Design*, ISBN 978-989-758-032-1, pages 196-201. DOI: 10.5220/0005426001960201.

Hong Li, Theodore J. Williams: *A Vision of Enterprise Integration Considerations - A holistic perspective as shown by the Purdue Enterprise Reference Architecture. ICEIMT/DIISM* 2004: 249-267.

Liqiang Chen 2012. *Integrating Cloud Computing Services Using Enterprise Service Bus (ESB), Business and Management Research*, ISSN 1927-6001 (Print) ISSN 1927-601X (Online).

Laleh Rafati. *Service Oriented Enterprise Engineering: Applying Viable System Approach (vSa) in Enterprise Engineering for Sourcing Decision Making. EDOC Workshops* 2013: 124-129.

- Markus, M. L., and C. Tanis (2000). *The Enterprise Systems Experience – From Adoption to Success*, in *Framing the Domains of IT Research: Glimpsing the Future through the Past*, R. W. Zmud (ed.), Cincinnati, OH: Pinnaflex Educational Resources Inc.
- Mumford, M.: *The story of socio-technical design: reflections on its successes, failures and potential*. Information Systems Journal 16(4), 317–342 (2006)
- OMG (2008). *Semantics of business vocabulary and business rules (sbvr)*, v1.0. Technical report, Object Management Group.
- RabahImache, SaidIzza, Mohamed Ahmed-Nacer: *An enterprise information system agility assessment model*. Comput. Sci. Inf. Syst. 9(1): 107-133 (2012).
- Ralf Schepres, Tobias Minning, Yannik Moog and Ingo J. Timm. *Towards Simulation of Business Processes Transformation BPMN Models to Enterprise Dynamics Models*. January 2014 fourth international Symposium on Business Modeling and Software Design BMSD 2014.

Metaprocesses as Software Assets Reuse

Javier Darío Fernández-Ledesma

*University Cooperative of Colombia, Department of Systems Engineering, calle 40 A N.41-05, Medellín, Colombia
javier.fernandez@ucc.edu.co*

Keywords: Metaprocess, Processes, Reuse.

Abstract: Software reuse in the early stages is a key issue in rapid development of applications. Recently, several methodologies have been proposed for the reuse of components, but mainly in code generation as artifacts. However, these methodologies partially consider the reuse of abstract models built for domain analysis and business modeling. This article introduces a metaprocess-oriented methodology based on the model reuse as software assets, and starting from the domain specification and analysis phases. The approach includes the definition of a conceptual level to adequately represent the domain and a reuse process to specify the metaprocess as software assets. The methodology has been applied successfully in the field of e-health; in particular a monitoring system for patients with cardiovascular risk has been modelled and analysed.

1 INTRODUCTION

Software reuse starting from the early stages of software development facilitates rapid development of applications. This contributes to increased productivity and quality in software development [8].

Metaprocess metamodeling and its specification as software assets for reuse in the early stages is a field in which much effort is needed to standardize software processes in the context of software industrialization.

The purpose of this article is to present a metamodeling-based approach intended to provide a conceptual perspective for a domain, and its specification as software assets to improve metaprocess reuse in early stages of software development. To illustrate the applicability of our proposal, this article also presents a real example currently running on the e-health domain: a monitoring system for patients with cardiovascular risk factors.

The remainder of the article is organized as follows. In section 2 we discuss related work. Section 3 provides a description to the proposed approach. Finally, we present the conclusions in Section 4.

2 RELATED WORK

There are several proposals regarding definition of metaprocesses in the context of software

development. A metaprocess can be defined as a complete process meta-model that serves as a benchmark to be instantiated, including different cases or situations for the same domain. This metamodel contributes to the generalization of processes through its metamodeling, specification of methods, decomposition of tasks, and rules of consistency. Rolland and Prakash (2000) conceive metaprocesses as artifacts with general features for instantiation, customization and gradual refinement of processes and models. Therefore, Metaprocesses are generic specifications of activities, tasks, roles, and behaviors supporting the execution of processes with the main objective of obtaining an abstraction of the domain.

There are several proposals regarding metaprocesses in the context of software development, such as metaprocesses as metamodeling-based models (Conradi and Nguyen, 1994) or methodologies for process-oriented software development (Finkelstein et al., 1994). However, none of them provide descriptions of the reuse of the metaprocess specification as software assets for instantiation and customization in the early stages.

Greenfield and Short (2004) define "[...] metaprocess-oriented methodology can contribute to raise productivity and quality in the software construction process ". In particular, a methodology that takes into account the abstraction of the domain specification and process modelling, as well as its specification as software assets for reuse in the early stages can ensure shorter development times before

the release of applications. This can also reduce costs in areas with strong demand for the development of rapid applications in domains such as e-health.

Nowadays, there are several approaches that deal with the use of metaprocesses, but, in general, reusability is not taken into consideration in the early stages of software development. Our objective is to consider metaprocesses, as part of a generic environment, that integrally support model-oriented software development, taking explicitly into consideration the specific domain.

Ramsin and Paige (2008) incorporate the use of metaprocesses as an oriented methodology towards model-driven software development. Their work describes what has been accomplished in terms of component reuse, but does not describe reuse of models from the early stages of software development. Ouyang et al. (2009) present a methodological proposal which is oriented to business processes models with BPMN and it translate to BPEL. Ouyang et al. (2009) Proposes a translation technique; but it don't consider the models reuse. Additional works have proposed a systematic approach to modelling that accompanies the use of metaprocesses in all phases of software development, starting from the early stages, in order to understand and analyse the domain, to design the software solution and build its implementation. Kühne (2009) incorporates the concept of metaprocesses into the evolution of software processes.

Asikainen and Männistö (2009) consider the need to semantically formalize software development processes using the metamodel processes. Levendovszky et al., (2009) incorporate the use of process metamodel patterns as a first step in formalizing specifications. However, these approaches do not take into account the issue of model reuse and its formalization in the early stages of software development; they use independent platform models and their implementation from the domain, but none of them include the problem of instantiation and customization through the reuse of metaprocesses.

Cechticky et al., (2006) proposes to reuse code components for real time applications. This proposal is based on the reuse of code, but does not cover models and metaprocesos. Park et al (2007), use code components as software assets to facilitate reuse of these components, which is done independently from models. De Freitas (2009) incorporates flexibility through the reuse of application code, without addressing the issues of model and metaprocesos reuse for generating applications. Finally, Rodriguez et al (2011) propose reusing design knowledge, in order to

reduce costs and difficulties in software development, as a contribution to the methods developed in collaborative systems such as groupware design patterns, pattern languages and frameworks. These other proposals do not include the theoretical and conceptual articulation of metaprocesos usage in software development through the fostering of reuse, instantiation and customization using platform independent architectures, and the use of models and metamodels as a contribution to software industrialization.

3 A METHODOLOGICAL PROPOSAL FOR THE REUSE OF METAPROCESSES

This section introduces a methodology that consists of one level design for the description of metaprocesos (conceptual level) and a reuse process (see Figure 1). The representation and construction of metaprocesos begin from a generic metamodel, which is specified using different models to fully represent the domain, with customization applied at different development stages, from specification of requirements to design and software applications. We seek for a sufficiently expressive and complete representation of metaprocesos in order to cover the conceptual domain with elements which facilitate the reuse in the software development.

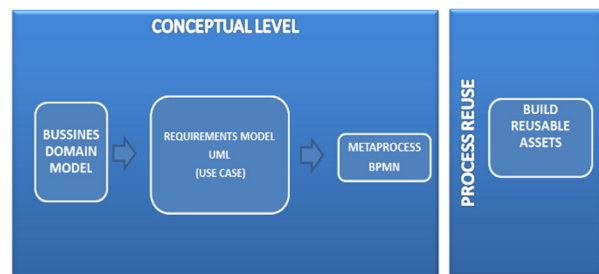


Figure 1: Metaprocesos specification levels as software assets expressed in models.

The conceptual level is a generic abstraction of the conceptual domain, which is represented through the business domain, requirements, and process models. The business domain model specifies the business cycle: mission, policies, business and process elements. The requirements model is based on use case diagrams. The metaprocesos is based on the use of the BPMN standard. A metaprocesos will be specified, as described below, on the basis of the business requirements captured through use cases and

process elements of the conceptual domain and their relationships.

Consequently, the conceptual level is specified using three steps as depicted in Figure 1:

1. The first step encompasses the representation of the domain through the Business Domain Model (BDM). BDM specifies business activities, business tasks and business roles. BDM contains diagrams such as the business diagram and process flow diagrams. The process flow diagram specifies the business activities and roles. In this way, a business activity is conducted using one or more activities process.
2. The second step is intended to build the Requirement Model (RM), including use case diagrams.
3. Finally, the third step is to build the Process Definition Model (PDM), which includes business process functionality both from a domain perspective and from the perspective of system support implementation for this software component.

This is followed by the metaprocess construction, based on information provided by the Domain Analysis, using the BPMN Notation, in which the activities (task in BPMN) and roles (swimlanes in BPMN) of the Metaprocess are clearly identified, as well as the use cases and applications or systems that support the execution of the metaprocess. Each activity and each role of metaprocess has been specified with a set of applications or systems or part thereof that support its execution for reuse in other cases as components reuse.

The reuse process enables the representation of metaprocesses as reusable software asset. Hence it represents the metaprocess architecture with its constituent elements specified as software artifacts. The specification of reusable software assets metaprocess as according to the standard is done by identifying each metaprocess as a reusable software asset with its component artifacts and attributes that describe them. It is defined by use, solution and classification, profile, related profiles and related assets as reusable software component elements of asset.

The reuse process is based on the OMG-AS and OMG's RAS standard, and uses a repository of reusable software asset to enable storage and search of assets in packed files, they contain these assets and an XML manifest (XML Schema) and, they are specified as .xsd as rasset.xml residing in the root directory of the asset accompanied by the respective. XSD or XML-Schema and another set of artifacts,

files or subdirectories that help specify. These components are compressed into a single file with a .Ras to facilitate management of its reusable software assets.

Finally, this propose uses the mechanisms for building well-formed models and Reusable Asset Specification (RAS).

The mechanisms for RAS facilitate the search and retrieval elements of the metaprocess as software assets (models, components, artifacts) into repositories. The OMG- RAS standard proposes the organization of the files .Ras and rasset.xml file structure, assets can be searched, retrieved and sail them through services, which can be implemented as Web Services or other approaches, which states for each service the nature of the response and the response.

In this case the implementation of mechanisms for RAS it's possible through our own repository *Actives*.

4 DISCUSSION AND CONCLUSIONS

This methodological approach contributes to specifying domains by means of conceptual levels. These levels facilitate the creation of design models independently from the platforms. In this manner, it is possible to obtain an understanding of the domain with the purpose of correcting problems inherited through deficient requirements gathering or a lack of comprehension of the same. As a result, we obtain specific elements at a conceptual level that can be reused in the development of future applications.

Our approach follows the Model-Driven Development (MDD) paradigm, through transformation of models between the early stages of the software development process.

The methodology proposed for the metaprocess at the conceptual specification level as software assets for reuse in the early stages of software development is intended to facilitate the development of domain process oriented applications, in this case for e-health. This methodology facilitates the software development process in one case, in which guided models contributed to the development of applications from the domain, independently from the development platforms. Now, models, metaprocesses, components and artifacts are being used to develop other systems, such as an interoperability platform for a pre-hospital system.

The monitoring system for patients with cardiovascular risk, and the interoperability platform

for the pre-hospital domain has been implemented at IPS University Hospital (Medellin, Colombia), and this system is being requested by other countries and other regional hospitals in Colombia.

The system has been evaluated by measuring its impact on the indicators. Briefly, the statistics and analysis of its implementation indicate that there is a significant improvement in the allocation of hospital resources and patient care times.

As future work, we plan to formalize the methodology through the use of logic languages using a formal definition Noguera et al. (2010), as well as considering the use of patterns and their mechanisms, for model reuse, as software as software assets not specified in the OMG standard.

ACKNOWLEDGEMENT

We thank DINAI of University Cooperative of Colombia.

REFERENCES

- Acuña, S. & Ferré, X. Software Process Modeling. pp.1-6 In: Proceedings of the 5th. World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001). Orlando Florida, USA. , (2001)
- Asikainen, T. & Männistö, T. Nivel: a metamodeling language with formal semantics, pp 521-549. *Software & Systems Modeling*. Volume 8, N. 4, (2009)
- Baisley, D., Björkander, M., Bock, C., Cook, S., Desfray, P., Dykman, N., Ek, A., Frankel, D., Gery, E., Haugen, Ø., Iyengar, S., Kobryn, C., Møller-Pedersen, B., Odell, J., Övergaard, G., Palmkvist, K., Ramackers, G., Rumbaugh, J., Selic, B., Weigert, T., Williams, L.: *OMG Unified Modeling Language (OMG UML), Superstructure v 2.2*. OMG (February 2009)
- Cechticky, V., Egli, M., Pasetti, A., Rohlik, O., Vardanega, T. A UML2 Profile for Reusable and Verifiable Software Components for Real-Time Applications, pp.312-325. *ICSR 2006, LNCS 4039*, (2006)
- Conradi, R. & Nguyen, M. Classification of Metaprocesses and their Models. pp 167-175. *Software Process*, (1994)
- De Freitas, J. Model business processes for flexibility and re-use : A component-oriented approach, pp 1-11. *IBM Developer Works Journal*, (2009)
- Finkelstein, A., Gabbay, D., Hunter, A., Kramer, J., Nuseibeh, B. *Software Process Modeling and Technology*, Research Studies Press LTD. Londres. (1994)
- Greenfield, J & Short, K. *Software Factories: Assembling Applications with Patterns, Model, Frameworks and Tools*. John Wiley & Sons. (2004)
- Kühne, T. Editorial to the theme issue on metamodeling, pp. 447-449. *Software & Systems Modeling*. Volume 8, N. 4, (2009)
- Levendovszky, T., László, L., Mészáros, T. Supporting domain-specific model patterns with metamodeling, pp 501-520. *Soft. & Systems Modeling*. Vol 8, N 4, (2009)
- Noguera, M., Hurtado, M., Rodríguez, M., Chung, L., Garrido, J. Ontology-driven analysis of UML-based collaborative processes using OWL-DL and CPN, pp 726-760. *Science of Computer Programming*. Volume 75, (2010)
- OMG. *Software & Systems Process Engineering Meta-Model Specification doc.ormsc/(2008-04-01)*
- OMG: *Business Process Model and Notation (BPMN) v1.2*. Object Management Group (OMG) (2008)
- OMG. *Reusable Asset Specification*. OMG Available Specification Version 2.2. (2005)
- Ouyang, C., Dumas, M., Van der aalst, W., Ter hofstede, A., Mendling, J. From business process models to process-oriented software systems. *CM Trans. Soft. Eng. Methodologies*. 19, 1, Art. 2 (August 2009)
- Park, S., Park, S., Sugumaran, V. Extending reusable asset specification to improve software reuse, pp 1473-1478. *Proceedings of the 2007 ACM symposium on Applied computing - SAC '07*, (2007)
- Ramsin, R. & Paige, R. Process-Centered Review of Object Oriented Software Development Methodologies. pp 1-89. *Computing*, Volume 40, N. 1, (2008)
- Rodríguez, J., Ochoa, S., Pino, J., Herskovic, V., Favela, J., Mejía, D., Morán, A. Towards a reference architecture for the design of mobile shared workspaces. pp 109-118. *Future Generation Comp. Systems*. Vol 27 (2011).
- Rolland, C., Prakash N. *On the Adequate Modeling of Business Process Families*. Université Paris1 Panthéon Sorbonne. Francia, (2000)
- Uribe, C., Isaza, C., Florez, J. *Qualitative-Fuzzy Decision Support System for Monitoring Patients with Cardiovascular Risk*, pp 1621-1625. *Proc. Conf. on Fuzzy Systems and Knowledge Discovery*. Vol. 3, (2011).

**SPECIAL SESSION ON
GREEN IT SOLUTIONS**

The Effect of Network Performance on High Energy Physics Computing

Jukka Kommeri^{1,2}, Aleksi Vartiainen^{1,2}, Seppo Heikkilä³ and Tapio Niemi³

¹*Helsinki Institute of Physics, Helsinki, Finland*

²*Aalto University, Espoo, Finland*

³*Helsinki Institute of Physics, CERN, Geneva, Switzerland*

{jukka.kommeri, aleksi.vartiainen}@aalto.fi, {seppo.heikkila, tapio.niemi}@cern.ch

Keywords: OpenStack, Network, Latency, Energy efficiency, Scientific Computing.

Abstract: High Energy Physics (HEP) data analysis consists of simulating and analysing events in particle physics. In order to understand physics phenomena, one must collect and go through a very large quantity of data generated by particle accelerators and software simulations. This data analysis can be done using the cloud computing paradigm in distributed computing environment where data and computation can be located in different, geographically distant, data centres. This adds complexity and overhead to networking. In this paper, we study how the networking solution and its performance affects the efficiency and energy consumption of HEP computing. Our results indicate that higher latency both prolongs the processing time and increases the energy consumption.

1 INTRODUCTION

High Energy Physics (HEP) studies elementary particles by using large particle accelerators, such as the Large Hadron Collider (LHC) at CERN, for producing millions of high-energy particle collision events. In order to understand physics phenomena, one must go through a very large quantity of measurement samples. A single high-energy physics analysis can process millions of events (Ponce and Hersch, 2004). This work can be easily parallelized because there are no dependencies among these events. Particle physics events are stored in database like containers, ROOT files (Antcheva et al., 2009). The required computing resources for CERN LHC data analysis are divided among 11 tier-1 sites and 155 tier-2 sites of computing centers world-wide using the grid/cloud computing paradigms (Bird et al., 2014). The distributed nature of HEP computing poses some extra overhead when the data needs to be accessed from a site that is geographically very distant. This often happens since the grid infrastructure used at CERN, World Wide LHC Computing Grid (WLCG), spans from Japan to USA. Although, WLCG was designed before the era of cloud computing, also different cloud solutions has been studied and the OpenStack cloud suite has been found suitable for HEP computing (Andrade et al., 2012; O’Luanaigh, 2014).

On the high level, cloud computing is a collec-

tion of servers, or hypervisors, that run mixed sets of virtual machines processing various workloads. The hypervisors share their processing and networking resources among a set of virtual machines. In the case of HEP data analysis jobs, which fetch constantly data from remote location, the network can become a bottleneck and cause delays for the analysis. The delays can have a big impact on overall performance.

ROOT files are most commonly accessed with the XRootD protocol, that runs on top of TCP, (Behrmann et al., 2010). The performance of XrootD is a well-studied topic. These studies mainly focus on storage performance (Gardner et al., 2014; Matsunaga et al., 2010), data federation (Bauerdick et al., 2014), and scalability (Dorigo et al., 2005; de Witt and Lahiff, 2014). Energy efficiency has not been considered, nor the effect of network delay on the performance of HEP computing. Therefore, in this paper we study the effect of networking in cloud environment on the performance and, especially, energy efficiency of HEP computing. In HEP the data and computing is geographically distributed all over the globe. For this reason the key problem examined in this paper is the performance of HEP software accessing locally and remotely located data sets. In particular, the goal is to understand the effect of latency and throughput to HEP job execution time and energy usage.

The remainder of this paper is structured as follows. First, in Section 2, we cover the related work.

In Section 3 we describe our test software and the test environment used in this study, which is followed by the results in Section 4. Then we end with conclusion in Section 5.

2 RELATED WORK

The importance of networking energy efficiency keeps growing as the world is getting more and more connected. Bolla et al. (Bolla et al., 2011) have studied what kind of research there has been in the network domain to improve energy efficiency. Ideas are very similar to those of the energy efficiency of computing. Hardware energy efficiency needs to be improved and the hardware needs better power scaling abilities: there needs to be a way to turn off hardware resources when they are not needed and in this way to improve the utilization level of the hardware. Kliazavich et al. (Kliazovich et al., 2010) have developed a cloud simulator, Greencloud, for measuring the energy consumption of cloud data centers. It can, e.g., evaluate different internal network layouts. The authors also showed how cloud network can benefit from load based management of virtual machines.

Load based management of virtualized cluster has been studied a lot and many different algorithms with various heuristics have been proposed. Piao et al. (Piao and Yan, 2010) have studied this topic from networking aspect. They have developed an algorithm that moves virtual machines in order to avoid congestion in the network. In their simulations, the execution time of data intensive application was improved by up to 25% when using their network traffic aware algorithm. Kuo et al. (Kuo et al., 2014) have studied how internal cloud latencies affect MapReduce (Dean and Ghemawat, 2008) performance. They have developed a virtual machine placement algorithm that attempts to minimize the network latency between cloud instances and this way improve the performance of Hadoop tasks.

In a shared environment like cloud, also the internal networking can become a bottleneck. Mauch et al. (Mauch et al., 2013) introduce High Performance Cloud Computing (HPC2) model. They have studied how suitable Amazon cloud would be for HPC computation and found the network to be limiting factor for performance. The 10Gb network of Amazon was found to cause more than ten times more latency than Infiniband that is normally used in HPC clusters. Exposito et al. (Exposito et al., 2013) have also studied how well different HPC loads perform in Amazon cloud. Reano et al. (Reano et al., 2013) have found similar limitations for remote GPU computing.

The Gigabit Ethernet can add a 100% overhead on rCUDA¹.

As the physics analysis jobs are well parallelizable and do not need inter process communication, the performance of internal communication is not so important. More important is the access time to data, which depends partly on cloud internal networks, but also on how far the data is and how the cloud connects to it. Haeussler et al. (Haeussler et al., 2015) have studied how latency effects the performance of genome annotation data retrieval. In their case, the data can be very far away and this distance can slow down the retrieval process significantly. Shea et al (Shea et al., 2014) have studied the performance of TCP in cloud environment. They have shown that the network performance of virtual machine depends on the CPU load of its hosting hypervisor, i.e., if there are other virtual machines on the same physical host with high CPU load, there is less CPU time for networking.

The same situation occurs when virtual machine itself has a high CPU load. This study was made in Amazon environment and with a separate Xen setup. Bulot et al. (Bulot et al., 2003) have studied the performance of different TCP variants. They have compared different TCP versions over connections that link continents. The distance affects the performance of different TCP variations differently. Scientific computing runs mainly on Linux machine and CERN has its own variant of it, Scientific Linux at CERN (SLC). SLC uses Cubic TCP, which is a default TCP variant in Linux and a more fair version of BIC TCP (Ha et al., 2008).

As the review above shows, many aspects of networking of cloud clusters, data centers, and the Internet have received a lot of research attention, still the effect of latency on energy efficiency of different workloads have not been much studied.

3 TEST ENVIRONMENT

There are different kinds of HEP workloads: simulation, reconstruction, analysis, etc. In this study, we used as our workload a process that transforms real physics event data into a more compact form that can be eventually used by the physicist on a standard PC hardware. The transformation process of a single event has two phases. In the first phase, the events are selected based on their suitability for the current analysis. Then, in the second phase, the event data is transformed and stored in a more space saving structure.

¹<http://www.rcuda.net/>

HEP computation uses special software packages. In the case of CERN CMS experiment, CMS software framework (CMSSW) (Fabozzi et al., 2008) is used. CMSSW is distributed to computing nodes with CERN Virtual Machine Filesystem, CVMFS (Meusel et al., 2015). CVMFS is a centrally managed software repository that contains several versions of various HEP software frameworks. It can be mounted directly to computing nodes. The software is cached locally when it is being used. In the case of CMS analysis, the job can cache about 1 GB of data or program code.

In this study, we tested HEP workload in a cloud environment. Since the location of the data and networking conditions have a big impact to the performance of the computation, our goal is to measure this effect of distance on both computation time and energy consumption. In our tests, the OpenStack² cloud platform was used. OpenStack is a open-source platform for cloud computing consisting of individual projects, i.e. services, that are responsible for computing, networking and storage, among other services. OpenStack services are designed to be deployed on multiple nodes, with a scalable number of compute nodes. Virtual machines running on OpenStack are called instances.

Measurements were done using three different cloud setups. In all the setups, we had a HEP client that reads data from storage server and does the transformation. In every test, the client was run in a cloud instance and the data server on a separate instance or on a separate server outside the cloud. The same workload was used in all the tests and its run times and energy consumption in different network conditions were measured. Tests were repeated several times to get reliable results.

3.1 Local Data

First tests were done using a single physical server with suitable hardware for energy measurement. This single server OpenStack cloud installation was setup with DevStack³. In our DevStack installation, all of the OpenStack components run on the same hardware. The setup used the following hardware: Fujitsu Esprimo Q910 computer with quad-core Intel(R) Core(TM) i5-3470T CPU @ 2.90GHz, 8 GB of RAM and 8 GB of swap. As an operating system, it had Ubuntu 12.04.

Physics workload was run in an OpenStack instance. OpenStack instances can have different amounts of resources; number of virtual processors

(VCPU), the amount of memory (RAM), and the size root disk and swap disk. These different configurations are called flavors⁴. Two types of flavors were used in the test environment: when running only one job the instance was assigned two virtual CPUs, 4 GB of memory and two GB of swap, and when running two jobs it was assigned one virtual CPU, 3 GB of memory and 1 GB of swap.

As a storage server for ROOT files, we used a ProLiant BL280c G6 blade computer with 16-core Intel Xeon CPU E5640 @ 2.67GHz and 68 GB of RAM. The server was in the same local area network of computer science department of Aalto University as the cloud setup and were initially connected with 100Mbe, which was upgraded to 1GbE for comparison. The server was installed with Ubuntu 14.04 and xrd server version 4.1.3.

We measured aspects such as processor (CPU), memory (RAM), power, cached data and network traffic statistics. Power measurements were done using Running Average Power Limit (RAPL) (Hähnel et al., 2012). RAPL is an Intel technology that measures the power consumption in Sandy Bridge CPUs and above. Network traffic has been recorded with Tshark, which is the command line version of Wireshark⁵ packet analyzer.

The workload was run in varying conditions, including network delay, packet loss, packet duplication, packet corruption, limited network throughput, parallel jobs, and different operating system cache and CVMFS cache configurations. In this paper, we use the term throughput to describe the actual network transport capacity, i.e., bits per second. The network limitations were simulated using the classless queuing disciplines (qdisc) tool, except for limited throughput simulated with Wondershaper⁶. The OS cache was cleared by freeing pagecache, dentries and inodes, and CVMFS cache with CVMFS tool `cvmfs_config wipecache`.

In addition to previously described single node cloud system, we installed a separate cloud, which was able to run more virtual machines, but lack the ability to measure energy consumption. We used the same blade hardware and operating system as previously for storage server. In these tests, we used three blades, which were installed using Puppetlabs OpenStack module⁷. The OpenStack controller node, networking node and compute node were installed on their own hardware. All the nodes were connected

⁴<http://docs.openstack.org/openstack-ops/content/flavors.html>

⁵<https://www.wireshark.org/>

⁶<http://www.lartc.org/wondershaper/>

⁷<https://github.com/puppetlabs/puppetlabs-openstack>

²<https://www.openstack.org/software>

³<http://docs.openstack.org/developer/devstack/>

to the same gigabit network switch. Data was served from the same node as where the controller was installed, but not within OpenStack. Physics analysis was run in an OpenStack instance. Tests with this cloud setup were done using varying amounts of virtual machines. In similar way as in single node tests, latency was simulated using qdisc.

3.2 Remote Data

Two OpenStack (release 2015.1.2) installations were used in this study. One was located in CERN (Meyrin, Switzerland) and another in the Aalto University (Espoo, Finland). Both of the deployments used three physical machines. The physical machines used in CERN were Dell PowerEdge R210 rack servers and in Aalto HP blade servers in HP BladeSystem c7000 Enclosure. The roles of these three machines were computing, networking and other services. Power consumption was measured only over the physical machine running computing service. Both of the OpenStack instances were configured with routable IP addresses in order to be accessible from outside.

The OpenStack installation in CERN was used to run three identical computing jobs in parallel. Each job was run in a separate Virtual Machine (VM). The jobs accessed 1.4GB physics data file hosted in XRootD servers. Two of these XRootD servers were hosted inside the Aalto and CERN OpenStack installations. Two other XRootD servers were deployed to existing OpenStack VMs: one in the CERN IT department and one to Kajaani in Finland.

In this study duration of HEP job processing and energy usage were collected. The duration was measured from the start of job processing until first VM finished the processing. Energy usage was collected during this same period of time.

4 RESULTS

We used three different testbeds to get diverse measurements. The workload was the same in every testbed. Depending on the setups, energy was measured either with an energy meter or by calculating from processor energy counters.

4.1 Local Data

Running the workload is both CPU and network intense. Figure 1 shows the relation between power consumption and network traffic, both appearing in synchronous cycles. The base power consumption of the hypervisor is typically less than five watts. Results

have a 30-second period of time in both ends when the virtual machine is running idle, i.e., no workload. This demonstrates the base power consumption and other base statistics on the hypervisor and the virtual machines. Remote resources from the XRootD server are downloaded in distinct parts. Most of the time there is no traffic between the server and the hypervisor. Closer look at the network throughput peaks show that there is short but constant peak that uses all the available bandwidth.

Network delays are typical to wide area networks (WAN) and have a clear impact on the workload runtime. As Figure 2 shows, an increase of 75 milliseconds in network delay, causes the run time to increase by over ten seconds. In addition, the power consumption increases by 34 percent. The effect is less evident when the same test is repeated in a network with more bandwidth. In Figure 3, we have the results of the same measurements in gigabit Ethernet. Run times are shorter with 1GbE, but energy consumption depends on latency. The effect of bandwidth is summarized in Table 1. In a gigabit LAN, the run times decrease roughly by ten percent when compared to that of a 100 Mbps network. Latency does not seem to have a big impact on the energy consumption when using 1GbE network, but an impact on 100MbE network.

Table 1: Comparison of execution times in different networks with no added delay.

Parallelism	Execution time (s)		Δ
	100MB/s	1 GB/s	
1 VM	281	255	-9 %
2 jobs, 1 VM	363	327	-10 %
2 jobs, 2 VMs	315	278	-12 %

The workload downloads and stores roughly 400 MB of data in CERN VM File System (CVMFS) local cache. If the cache is empty, this data is downloaded from CERN servers at the beginning of executing the workload. Otherwise no data exceeding ten kB in total is downloaded from CERN. As the total size of required tools is 400 MB, setting the CVMFS cache limit lower than that, affects the processing time and network traffic. As Table 2 shows, no data needs to be downloaded if the cache limit is high enough and the data has been previously cached. On the contrary, the workload cannot be executed at all if the cache limit is too low.

In Table 3, we have a summary of parallel workload tests. It shows that the latency does not increase significantly even though ten virtual machines are sharing a single physical interface. Flows have different latencies depending on the direction, but this

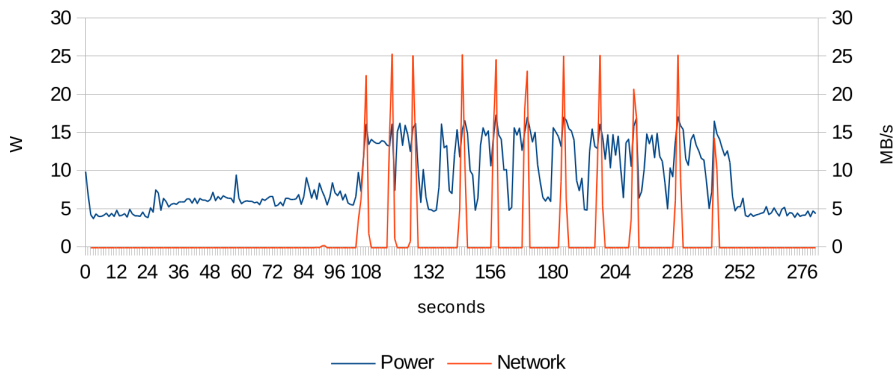


Figure 1: Relation between power consumption and network traffic on the hypervisor.

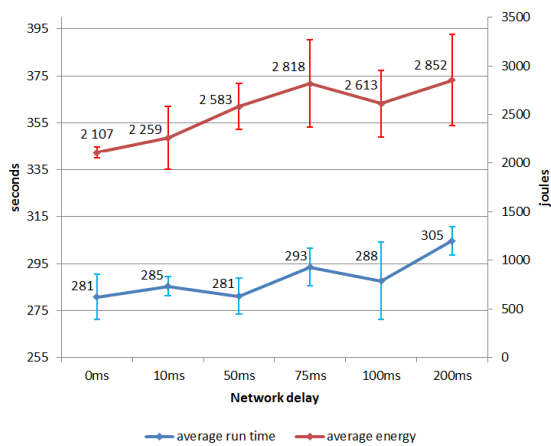


Figure 2: Comparison of workload run times and total energy consumptions with different network delays in 100Mb/s Ethernet.

Table 2: Comparison of execution times and network traffic from CERN. When the cache limit is sufficiently less than 300 MB, the workload cannot be executed.

* First run after CVMFS cache clear
 ** The following runs (average)

Cache limit	Data from CERN (MB) *	**	Execution time (s) **
200 MB	-	-	-
300 MB	422	300	249
500 MB	315	0	166
5000 MB	320	0	165

is similar with all workloads. Some change in the maximum values, but means and medians are about the same.

We tried to stress the shared interface even more by adding five additional virtual machines generating HTTP traffic by downloading large images from a university server. The results of this addition were similar to the results of 10VM workload.

Similarly to previous single server measurements,

Table 3: Round trip times between XRootD server and OpenStack instance in milliseconds.

	1VM	10VM
min	0.11	0.03
max	238.97	294.71
median	4.38	4.46
mean	7.67	10.14
stdev	25.32	32.08

we tested how added latency affects the execution time when running multiple virtual machines in parallel. Figure 4 shows how execution times increase when we add more latency. This measurement was done with 1, 5 and 10 virtual machines. The effect of added latency was greater with 1VM where 100ms caused 6.9% increase to execution time as with 10VMs it is 2.3%.

From the same 10VM tests we got the throughput values, that are shown in Figure 5. These results show a relation between latency and throughput as the maximum throughput decreased 41% when 100ms latency was added.

4.2 Remote Data

Figure 6 and Figure 7 show job run times and energy usage, respectively, when data is hosted with XRootD server in different physical locations. The energy usage is directly related to the processing time. Only the OpenStack energy usage is slightly higher because the measurement includes also the XRootD server hosting. Thus, it would be possible to estimate the energy usage by measuring only the processing time.

There is at least two possible causes for the differences between sites: the network latency and throughput. The network latencies to Aalto and Kajaani are 46.1 ± 0.3 ms and 49.6 ± 0.2 ms, respectively, while inside CERN the latencies are less than one millisecond. The network throughput to Aalto and Kajaani are 23.2 ± 2.7 MB/s and 20.0 ± 1.6 MB/s, respectively, while inside CERN 58.3 ± 4.9 MB/s.

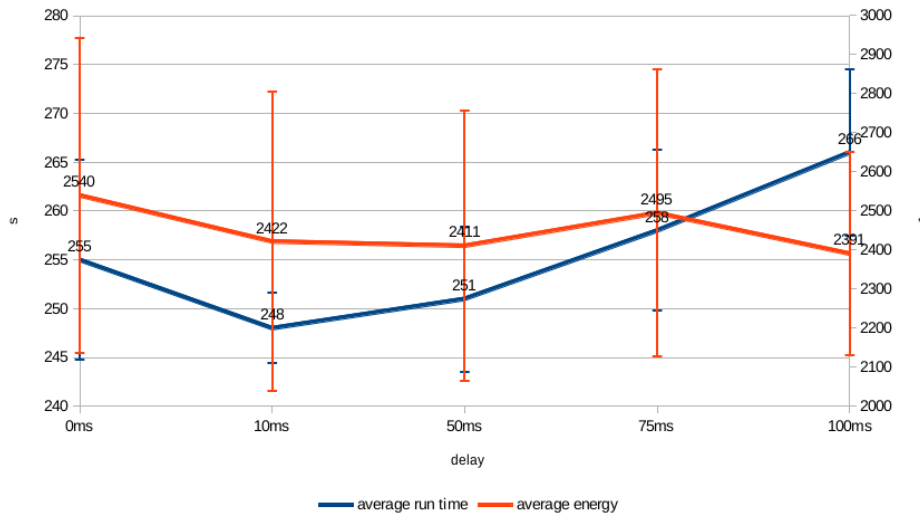


Figure 3: Comparison of workload run times and total energy consumptions with different network delays in 1Gb/s Ethernet.

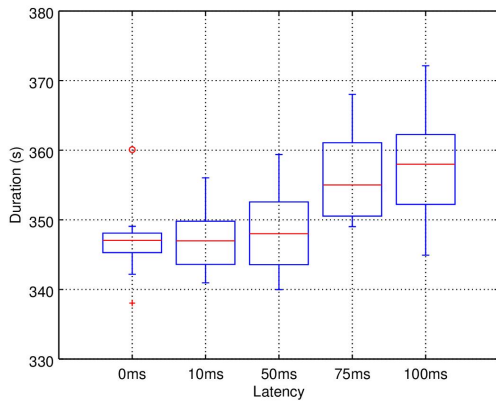


Figure 4: Job execution times with ten OpenStack instances in parallel and with added latency.

The effect of network throughput and latency can be examined by comparing local and remote sites with a 20MB/s fixed throughput that is achievable with both sites. Figure 8 shows that there is around 12.5 percent difference between CERN and throughput limited Kajaani site. Throughput limitation removes 70 percent of this difference so effect of latency seems to explain around 30 percent of the difference. A detailed analysis of the network traffic showed that only 0.01% TCP packets were retransmitted and also the TCP window size increased quickly to around three MB. Thus network problems do not explain the observed differences. With the current HEP software stack the best option is to prefer nearby data sources with low latency.

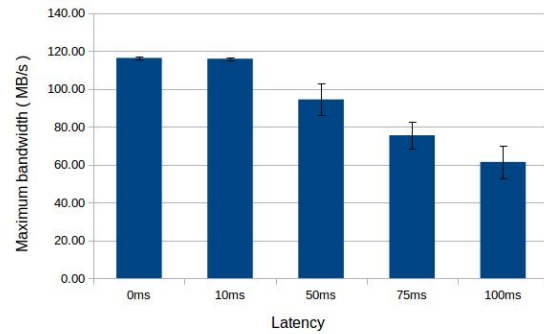


Figure 5: Maximum throughput with ten OpenStack instances in parallel and with added latency.

5 CONCLUSIONS AND FUTURE WORK

High energy physics computing at CERN uses a large computing grid/cloud distributed around the world. This naturally poses long distances between the sites and slows down the network connections among them. To alleviate this, we studied how networking performance affects on computing performance and energy efficiency on high energy physics computing in an OpenStack cloud testbed. We used both simulated network latencies in laboratory network and several geographically distant sites connected by the Internet to measure how different latencies change computing performance when processing HEP workload.

Our results indicate that the network latency, either caused by a simulator or physical distances between the sites, has a negative impact on the com-

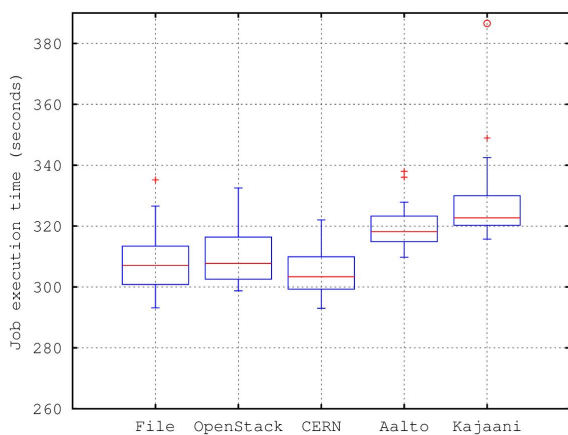


Figure 6: Job execution times of OpenStack VMs.

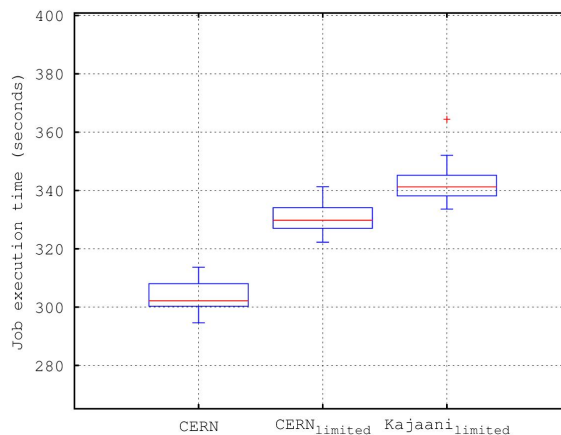


Figure 8: Job execution times with throughput limits.

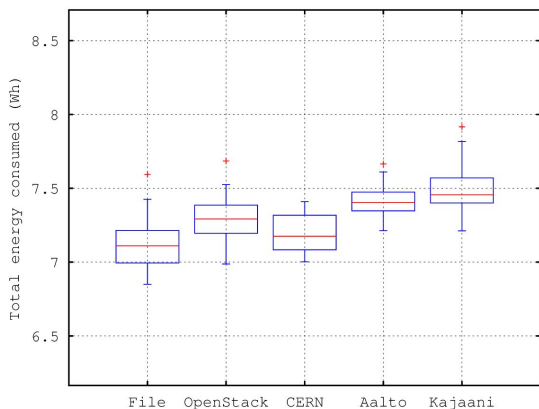


Figure 7: Energy usage of OpenStack VMs.

puting performance. High latency both increases run times and the total energy consumption. Additionally, we also noticed that the contribution of latency, to the execution time and energy consumption of a computation job, increases when bandwidth is small. Parallelism, multiple cloud instances sharing the limited network resource, also adds more latency and increases job run times.

The obtained results reflect the current software environment used for HEP job processing. New data transfer protocols or advanced caching mechanism could diminish the observed differences. Instead, the used network infrastructure and computing hardware is unlikely to change significantly in the near future.

Our future work includes studying methods how the effect of latency can be minimized using e.g. smarter workload scheduling, data preloading, or optimized network protocols.

ACKNOWLEDGMENTS

This paper has received funding from the European Union's Horizon 2020 research and innovation program 2014-2018 under grant agreement No. 644866.

DISCLAIMER

This paper reflects only the authors' views and the European Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

- Andrade, P., Bell, T., van Eldik, J., McCance, G., Panzer-Steindel, B., dos Santos, M. C., Traylen, S., , and Schwickerath, U. (2012). Review of cern data centre infrastructure. *Journal of Physics: Conference Series*, 396(4).
- Antcheva, I., Ballintijn, M., Bellenot, B., and Biskup, M. (2009). ROOT - A C++ framework for petabyte data storage, statistical analysis and visualization. *Computer Physics Communications*, 180(12):2499–2512.
- Bauerdick, L. A. T., Bloom, K., Bockelman, B., Bradley, D. C., Dasu, S., Dost, J. M., Sfiligoi, I., Tadel, A., Tadel, M., Wuerthwein, F., Yagil, A., and the Cms collaboration (2014). Xrootd, disk-based, caching proxy for optimization of data access, data placement and data replication. *Journal of Physics: Conference Series*, 513(4).
- Behrmann, G., Ozerov, D., and Zanger, T. (2010). Xrootd in dcache - design and experiences. In *International Conference on Computing in High Energy and Nuclear Physics (CHEP)*.
- Bird, I., Buncic, P., Carminati, F., Cattaneo, M., Clarke, P., Fisk, I., Girone, M., Harvey, J., Kersevan, B., Mato,

- P., Mount, R., and Panzer-Steindel, B. (2014). Update of the computing models of the wlcg and the lhc experiments. Technical report, CERN.
- Bolla, R., Bruschi, R., Davoli, F., and Cucchietti, F. (2011). Energy efficiency in the future internet: A survey of existing approaches and trends in energy-aware fixed network infrastructures. *Communications Surveys Tutorials, IEEE*, 13(2):223–244.
- Bulot, H., Les Cottrell, R., and Hughes-Jones, R. (2003). Evaluation of advanced tcp stacks on fast long-distance production networks. *Journal of Grid Computing*, 1(4):345–359.
- de Witt, S. and Lahiff, A. (2014). Quantifying xrootd scalability and overheads. *Journal of Physics: Conference Series*, 513(3).
- Dean, J. and Ghemawat, S. (2008). Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113.
- Dorigo, A., Elmer, P., Furano, F., and Hanushevsky, A. (2005). Xrootd-a highly scalable architecture for data access. *WSEAS Transactions on Computers*, 1(4.3).
- Expsito, R. R., Taboada, G. L., Ramos, S., Tourio, J., and Doallo, R. (2013). Performance analysis of HPC applications in the cloud. *Future Generation Computer Systems*, 29(1):218 – 229. Including Special section: AIRCC-NetCoM 2009 and Special section: Clouds and Service-Oriented Architectures.
- Fabozzi, F., Jones, C., Hegner, B., and Lista, L. (2008). Physics analysis tools for the cms experiment at lhc. *Nuclear Science, IEEE Transactions on*, 55:3539–3543.
- Gardner, R., Campana, S., Duceck, G., Elmsheuser, J., Hanushevsky, A., Hönig, F. G., Iven, J., Legger, F., Vukotic, I., Yang, W., and the Atlas Collaboration (2014). Data federation strategies for atlas using xrootd. *Journal of Physics: Conference Series*, 513(4):042049.
- Ha, S., Rhee, I., and Xu, L. (2008). Cubic: A new tcp-friendly high-speed tcp variant. *SIGOPS Oper. Syst. Rev.*, 42(5):64–74.
- Haeussler, M., Raney, B. J., Hinrichs, A. S., Clawson, H., Zweig, A. S., Karolchik, D., Casper, J., Speir, M. L., Haussler, D., and Kent, W. J. (2015). Navigating protected genomics data with ucsc genome browser in a box. *Bioinformatics*, 31(5):764–766.
- Hähnel, M., Döbel, B., Völp, M., and Härtig, H. (2012). Measuring energy consumption for short code paths using rapl. *ACM SIGMETRICS Performance Evaluation Review*, 40(3):13–17.
- Kliazovich, D., Bouvry, P., and Khan, S. U. (2010). Green-cloud: a packet-level simulator of energy-aware cloud computing data centers. *The Journal of Supercomputing*, 62(3):1263–1283.
- Kuo, J.-J., Yang, H.-H., and Tsai, M.-J. (2014). Optimal approximation algorithm of virtual machine placement for data latency minimization in cloud systems. In *INFOCOM, 2014 Proceedings IEEE*, pages 1303–1311.
- Matsunaga, H., Isobe, T., Mashimo, T., Sakamoto, H., and Ueda, I. (2010). *Managed Grids and Cloud Systems in the Asia-Pacific Research Community*, chapter Performance of a disk storage system at a Tier-2 site, pages 85–97. Springer US, Boston, MA.
- Mauch, V., Kunze, M., and Hillenbrand, M. (2013). High performance cloud computing. *Future Generation Computer Systems*, 29(6):1408 – 1416.
- Meusel, R., Blomer, J., Buncic, P., Ganis, G., and Heikkilä, S. S. (2015). Recent developments in the cernvm-file system server backend. *Journal of Physics: Conference Series*, 608(1):012031.
- O’Luanaigh, C. (2014). Openstack boosts tier 0 for lhc run 2. Technical report, CERN.
- Piao, J. T. and Yan, J. (2010). A network-aware virtual machine placement and migration approach in cloud computing. In *Grid and Cooperative Computing (GCC), 2010 9th International Conference on*, pages 87–92.
- Ponce, S. and Hersch, R. D. (2004). Parallelization and scheduling of data intensive particle physics analysis jobs on clusters of pcs. In *18th International Parallel and Distributed Processing Symposium (IPDPS 2004)*, Santa Fe, New Mexico, USA.
- Reano, C., Mayo, R., Quintana-Orti, E., Silla, F., Duato, J., and Pena, A. (2013). Influence of infiniband fdr on the performance of remote gpu virtualization. In *IEEE International Conference on Cluster Computing (CLUSTER)*, pages 1–8.
- Shea, R., Wang, F., Wang, H., and Liu, J. (2014). A deep investigation into network performance in virtual machine based cloud environments. In *Proceedings of IEEE INFOCOM*, pages 1285–1293.

A Backpressure Framework Applied to Road Traffic Routing for Electric Vehicles

Evangelos D. Spyrou and Dimitrios K. Mitrakos

*School of Electrical and Computer Engineering, Aristotle University of Thessaloniki,
Egnatia Odos, Panepistimioupoli, Thessaloniki, Greece
{evang_spyrou, mitrakos}@eng.auth.gr*

Keywords: Electric Vehicles, Routing, Backpressure Weight, Lyapunov.

Abstract: Electric vehicles (EVs) emerged in the transport domain, due to their energy efficiency and clean energy that they utilise. The electric vehicle routing problem is essentially a problem of selecting a set of minimum cost routes, while the demand of the customers is achieved. Route cost metrics include energy consumption and driving time. In this work, we model the electric vehicle routing problem using a wireless network methodology, namely the backpressure framework. The penalty imposed to every route includes the driving time of each road. We derive a weight as a function of the road queue backpressure and the driving time of a car. The next route for our EV is the one that has the highest weight. It turns out that this methodology leads to faster routes in that there are often roads with accidents or traffic jams, even though they are in the shortest path of the route to the destination. We present results via simulations, which verify the fact that backpressure is an efficient algorithm to be applied to electric vehicle routing.

1 INTRODUCTION

The use of electric vehicles (EVs) into the transport sector has been introduced (Emadi, 2011) for two reasons. Firstly, there is the need for energy efficiency in transport, since the charging of the EV is not expensive as opposed to the internal combustion engine vehicles. Secondly, there is the reduction of the CO_2 emissions due to the clean sources (battery) employed for the production of electricity. On the other hand, traditional vehicles are associated to the combustion of fossil fuels. Thus, EVs provide a great alternative as a next generation of the transport means in a city.

The main characteristics of EVs play a key role on the design of algorithms utilised by route planners. EV route planning exhibits certain differences from conventional route planning. Initially, the limited capacity of the EV's battery introduces the constraints of driving ranges of approximately 100 km on average. Moreover, charging of EVs take place in stations that do not exist in as many places as gas stations. EV charging is a process that may take hours to complete. Thus, routes may be determined in an economic manner rather than just the fastest or shortest. On the other hand, there are cases where the state of the battery does not necessarily decrease when driving. All of the above sum up to the fact that companies with

EV fleets, require their goods to be delivered on time in often jammed road networks with potential uncertainties, such as road accidents or road works.

The Vehicle Routing Problem (VRP) was proposed in (Dantzig and Ramser, 1959). Thereafter, a plethora of extensions of the problem have been suggested, which included real world constraints. Two works that constitute the most widely investigated extensions are the Capacitated VRP (CVRP), where vehicles have a limited freight capacity and the VRP with Time Windows (VRPTW), where customers have to be reached within a specified time interval (Laporte, 2009) (Nagata et al., 2010). The Electric Vehicle Routing Problem (EVRP) (Touati-Moungla and Jost, 2012; Artmeier et al., 2010) has been formulated as a problem of locating a set of minimum cost routes, in order for the demand of the customer to be accomplished. Often, constraints related to the capacity of the battery of the EV have been investigated.

In this paper we deal with dynamic traffic routing for electric vehicles. For our approach, we employ a methodology utilised in wireless networks, called the Backpressure routing, which essentially builds the routing without routes (Moeller et al., 2010). The aim is to employ infrastructure, such as traffic cameras to estimate the queues created at each traffic light road.

This will provide the driving time of each street and we provide a penalty minimisation used in a weight to be calculated for each road. We provide results based on simulations to show the efficiency of our approach, which results in the successful delivery of companies' goods on a predefined schedule in a single depot. We show the following:

- EV routing may be modelled with the backpressure framework, which exhibits stable queues and they are used to construct a weight to select the next best route
- Road driving time is a metric that may act as a penalty that is minimised.
- Backpressure application results in fastest routes, avoiding any sudden road condition changes in a real-time fashion.
- There are cases that backpressure outperforms Dijkstra's shortest path algorithm in reaching a destination via EV routing.

The paper is structured as follows: Section 2 presents the related work in electric vehicle routing, section 3 provides a description of the backpressure framework utilised, section 4 explains the penalty to be minimised, section 5 gives the simulations of our proposed scheme and section 6 gives the conclusions of our approach.

2 RELATED WORK

In (Afroditi et al., 2014), the authors investigate the one-to-many vehicle routing and scheduling problem in EVs. More specifically, problem formulation and constraints in practical scenarios are examined. They highlight that the EVRP is an NP-hard problem and requires significant computational power for the location of near optimal solutions in medium to large scale scenarios. The authors provide a mathematical formulation to model the EVRP due to its capacity, time window and predefined charging level constraints. Furthermore, EVRP trends are examined providing significant information regarding future research on real world scenarios and approximation algorithms.

In (Bruglieri et al., 2015), the authors aim to find the optimal route for EVs in a multi-customer scenario considering recharging requirements during the routes. They formulate routing as a Mixed Integer Linear Programming problem. The battery recharging at every station is a variable, since flexible routes are to be guaranteed. The proposed scheme optimises total travel, waiting and recharging time as well

as the number of the EVs utilised. The problem is solved using Variable Neighbourhood Search Branching (VNSB) in reasonable computational times.

In (Schneider et al., 2014), the authors introduce the electric vehicle-routing problem with time windows and recharging stations (E-VRPTW). The model employs recharging at any of the predetermined stations using a recharging scheme. Furthermore, they consider limited vehicle freight capacities in conjunction with customer time windows, which constitute constraints in real-world transport applications. To solve the aforementioned problem, a hybrid heuristic is presented, which combines a variable neighbourhood search algorithm with a tabu search heuristic.

In (de Weerd et al., 2015) the authors propose an intention-aware routing system (IARS) for electric vehicles. The system provides the ability to EVs to estimate a routing policy, which minimises journey time, while keeping track of other vehicles intentions. Considering other vehicles' intentions is significant since the driver may have to charge the vehicle in the journey and queueing time may be large, in case other vehicles select the same stations. Thus, queueing times are predicted based on the intentions of the other EVs.

In (Abousleiman and Rawashdeh, 2014), the authors attempt to tackle the problem of energy efficient routing for EVs using Particle Swarm Optimisation. They also show that EVs route optimization techniques, such as negative edge costs, battery power and capacity limits, as well as vehicle parameters that are only available at query time, make the task of electric vehicle routing a challenging problem.

In (Baum et al., 2014) the authors investigate route planning applications for electric vehicles. They show that such problems have to consider constraints such as energy consumption. They indicate that recent approaches for EV routing focus on optimizing energy consumption as a single variable. They provide preliminary work towards a holistic framework for computing shortest paths for electric vehicles with limited range. Their scheme comprises driving energy-efficient speed adjustments, realistic modeling of battery charging and the integration of turn costs.

3 BACKPRESSURE FRAMEWORK

The dynamic nature of the road network combined with the unwanted but sometimes occurring bottlenecks, provide the necessary means for the emergence of a dynamic routing algorithm, where routing deci-

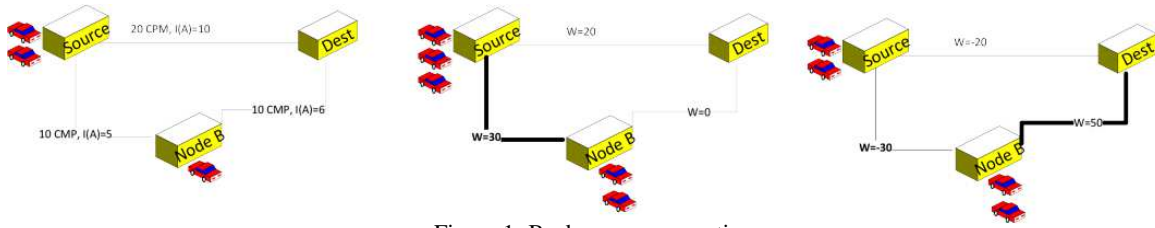


Figure 1: Backpressure operation.

sions will be made dynamically each time a car enters a road leading to a junction. Hence, it is quite useful to employ a wireless network routing algorithm that performs this task, with respect to the road network conditions. Modern road networks utilise traffic cameras to monitor traffic and to prevent speeding above the limit. Hence, it is quite reasonable to assume that they may be used to calculate driving time and number of cars within a road. Furthermore, with the emergence of the Internet of Things (Kopetz, 2011) and Cyber-Physical Systems (Baheti and Gill, 2011), it is clear that infotainment systems in cars maybe connected to devices on junction traffic lights that may exchange traffic information, resulting thus, to a wireless road network. Hence it may be promising to employ a dynamic routing algorithm in a road network, if we consider it as a graph-theoretic entity (Bondy and Murty, 1976) where vertices are considered as junctions and edges as roads.

Backpressure routing does not operate like traditional routing mechanisms, meaning that it does not locate an explicit path estimation from any source to a destination. It performs routing decisions for each car by calculating for each outgoing road a backpressure weight. This weight is a function of localised queue and link state information. In figure 1 we observe the backpressure functionality in a simple road network. Note that *Node B* represents an intermediate routing junction.

We will next provide a definition of a stable network.

We denote the queue at junction i during time slot t as $Q_i(t)$. A network of queue backlogs is defined as strongly stable if:

$$\limsup_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}[Q_i(\tau)] < \infty, \forall i \quad (1)$$

Furthermore, we denote as $f(\vec{x}(t))$ the penalty, which comes as an outcome of routing decisions between queues in time slot t . We assume that f is non-negative, continuous, convex and entry-wise non-decreasing, meaning, $f(\vec{x}) \leq f(\vec{y})$ when $\vec{x} \leq \vec{y}$ entry-wise. Let $f(\vec{x})$ be the penalty value of the function that works on the average value of the \vec{x} vector. We formulate the road network as a stochastic optimi-

sation problem in which routing decisions minimise function f and keeping strongly stable queues simultaneously. That is:

$$\begin{aligned} \text{minimise: } & f(\vec{x}) \\ \text{subject to: } & \text{Strongly Stable} \end{aligned} \quad (2)$$

Assuming that $f(\vec{x})$ is a cost metric of the routing process, we can derive the solution of equation (2) using the Utility Optimal Lyapunov Networking framework (Neely et al., 2008; Neely and Ugaonkar, 2008). This can show that we may have the result of routing decisions resulting in a backpressure routing policy. More specifically, each junction computes the weight per outgoing road in every time-slot. The weight is given below:

$$w_{i,j} = (\Delta Q_{i,j} - \tau \theta_{i,j}) \overline{R}_{i,j} \quad (3)$$

where $\Delta Q_{i,j} = Q_i - Q_j$ is the queue backpressure and Q_i, Q_j are the backlogs of junctions i, j respectively, \overline{R} is the road car driving rate and $\theta_{i,j}$ a road usage penalty that depends upon the particulars of the utility and penalty functions of (2). The parameter τ is a constant trades system queue occupancy for minimising the penalty. We propose a decentralised approach where junction i calculates the backpressure weight of all its neighbouring junctions. Thereafter, it is used to determine independent routing. For example, junction i locates the road (i, j^*) , which has the highest value of the backpressure weight as the next route of the car. We assume, at this point, that the weight is always larger or equal to 0. In the case that the weights are equal, we adjusted the algorithm to select the route with the shortest distance.

4 PENALTY MINIMISATION

The main challenge we are facing is to identify the most suitable penalty function $f(\vec{x})$ in equation (2), in order to provide efficient performance when we use it in a real work road traffic network.

Initially, we give some preliminaries regarding the road traffic network problem. We model the road network using a directed graph $G(V, A)$, where V represents the junctions and A represents the roads. As

we mentioned at a previous section, we assume that each junction has traffic control cameras, or other devices that may monitor and calculate the rate and driving time of each vehicle. The graph includes two attributes on each road $a \in A$. The first is an estimate before hand for its driving time for the solution we request, and the second is the road performance function. We use the road performance function that has been utilised in (Jahn et al., 2005). In particular, the road performance function, I_α , maps the traffic rate x_α , to its driving time $I_\alpha(x_\alpha)$.

The road performance functions, which will serve as the penalty functions of (2) determine the impedance of roads for a variety of congestion levels. We wish our functions to be nondecreasing and differentiable, as well as $I_\alpha(x_\alpha)$ to be convex. In our experiments we employ the function derived by the U.S Bureau of Public roads (BUREAU, 1964)

$$I_\alpha(x_\alpha) := I_\alpha^0 \left(1 + \alpha \left(\frac{x_\alpha}{c_\alpha} \right)^\beta \right) \quad (4)$$

where I_α^0 represents the travel time under no congestion, $\alpha \geq 0$ and $\beta \geq 0$ are tuning parameters, and finally, c_α is the practical capacity (Patriksson, 1994).

We employ the metric (4) to be minimised in the backpressure framework and we have the following penalty function in (2):

$$f(\vec{z}) = \sum_i \sum_{j \in N_i} z_{i,j}(t) \bar{I}_\alpha \quad (5)$$

where N_i is the set of neighbouring junctions of junction i , I_α is the road performance and $z_{i,j}(t)$ is the number of routed cars over the road $i \rightarrow j$. Notably, $f(\vec{z})$ satisfies the properties of problem (2) and provides a backpressure weight, calculated by a junction i to a neighbour j as follows:

$$w_{i,j} = (\Delta Q_{i,j} - \tau \bar{I}_\alpha) \bar{x}_{i,j} \quad (6)$$

5 SIMULATIONS

We produced a road network in MATLAB consisting of 7 junctions that we assume they are traffic lights equipped with traffic cameras. Furthermore, we have 10 roads connecting the junctions in a fashion that is given in figure 2 (a). We also show the distance of the roads between the junctions. This is since we wish to compare our algorithm with Dijkstra's shortest path algorithm (Dijkstra, 1959).

The parameters that the backpressure algorithm uses are given in table 1. More specifically, we provide the times that a car requires to pass from a road

and the queue backpressure for each junction. Moreover, the rate of each road is assumed to be equal, notably 4 cars per round (traffic light green to red). Finally, we set the parameter $\tau = 2$. We assume that there are certain roads that require more time to be driven even though they are of longer distance, due to more cars selecting these routes or by an occurrence of accidents.

Table 1: Time and Queue Backpressure per Junction.

Road	Time	Queues (i,j)	Queue Backpressure
A-B	4	15-10	5
A-C	6	12-5	7
A-D	10	13-6	7
B-C	10	10-5	5
B-F	7	10-3	7
C-E	5	5-2	3
C-F	5	5-3	2
F-S	4	3-1	2
E-S	5	2-1	1
D-S	12	6-1	5

Initially, we run the algorithm computing the shortest path for a car starting from junction A to the destination S . We observe in figure 2 (b) that the route computed is $A \rightarrow C \rightarrow E \rightarrow S$. This can be easily seen from the distances provides for each road.

Thereafter, we applied the backpressure algorithm, where we see that the route selected by a car starting from the origin A , in order to reach the destination S is different from the shortest path algorithm. In figure 2 (c) we see that the car drives to junctions $A \rightarrow C \rightarrow F \rightarrow S$. Furthermore, the driving time of the route using the backpressure as opposed to the shortest path algorithm is shorter by 1 minute. We are positive that in other scenarios the time difference may be longer.

In figure 3 (a) - (c) we present the steps of the backpressure algorithm by showing the routes selected by cars in different junctions. Each vertex includes the letter of the junction and its respective queue size i.e., $A - 14$. Furthermore, we provide the backpressure weight in each edge of the graphs, in order to make the route selections clear. Finally, we converted the driving time from minutes to hours to calculate the backpressure weights. Note that we assume that the incoming cars to junction $A = 3$ and that the junctions' traffic lights work simultaneously, for the sake of simplicity.

During the first round that the traffic lights become green we have the following configuration, which appears in figure 3 (a). As we can see, the backpressure weights that are given in the edges of the figure, promote the cars to route from $A - C$, $B - C$, $C - E$, $D - S$ and $F - S$. Note that the routes to S occur, since there are no other routes from the respective junctions.

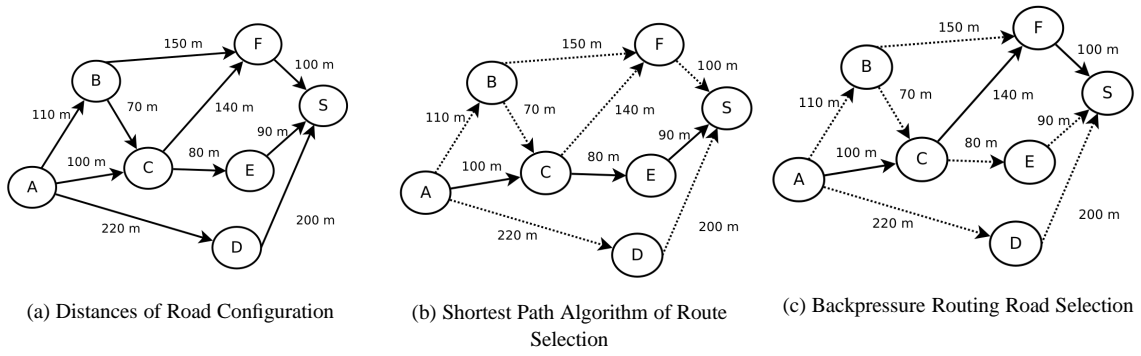


Figure 2: Routes of Backpressure, Shortest Path and initial configuration.

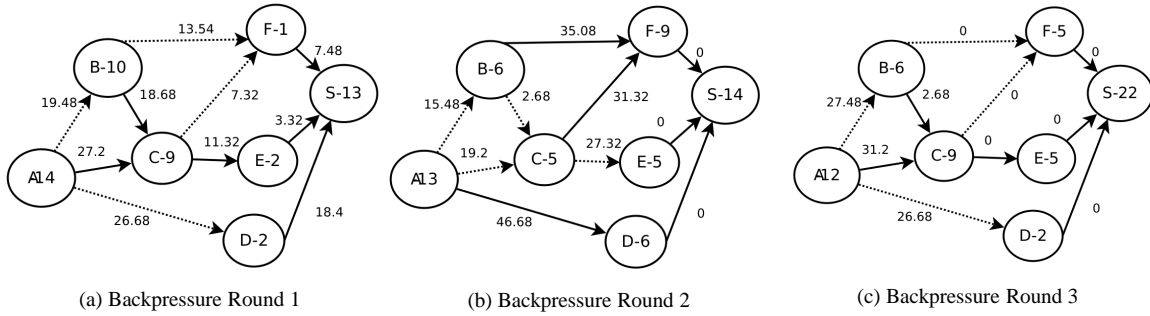


Figure 3: Backpressure Routing Rounds.

Furthermore, we assume that cars routed to junction C from junction A arrive first, since the driving time of the road is less than than the road originated from junction B.

In round 2 of the backpressure algorithm in figure 3 (b), we observe that the car that interests us moves from junction C to junction F, since the backpressure weight is higher than the one from C to E. Furthermore, the other routing decisions in this round consist of B – F, A – D and F, E, D to the destination S, since there are no other routes to select from. Moreover, we observe that these routes have a weight of 0, since it is the smallest value our algorithm may accept (the weight has a negative value in our calculations).

In figure 3 (c), the car that we are routing, reaches the destination S from junction F. We also see that other routing decisions include A – C, B – C, C – E and all the one hop junctions to the destination. Note that the routing decision for the routing of the cars of junction C to either F or E has been done using the shortest path, since both the respective backpressure weights are 0. The end of this round routes our car from junction A to the destination S.

6 CONCLUSIONS

In this paper we applied a state-of-the-art routing protocol for wireless networks to electric vehicle routing. The backpressure routing is essentially routing without routes. It calculates the next route of the car dynamically based on the calculation of the backpressure weight and the strong stability of the formed queues on the junctions. We utilised the driving time of each road as a penalty to be minimised by every car. The penalty is selected based on the necessity of a company’s fleet to reach the clients’ addresses within a given time window and in a sigle depot.

Our simulations showed that the backpressure algorithm creates faster routes than Dijkstra’s shortest path algorithm. This may be very useful, since roads may have several problems such as traffic accidents, road works or traffic jams, which may prolong their driving times even if the distance between two junctions is small. Hence the backpressure algorithm establishes an efficient routing mechanism.

Our future work includes the integration of the electric vehicle battery to the penalty minimisation process. This might provide a useful insight as to the route selection problem, since the fastest route may not be the most energy-efficient one. It is crucial

to compare our approach with state-of-the-art routing protocols for EVs that take into consideration charging station locations and charging times. The backpressure framework provides us with the flexibility to employ the penalty function of our liking, in order to produce the backpressure weight. Hence we are optimistic that it will perform well comparing other routing schemes. Moreover, our approach is not restricted to EVs only and it may be adjusted to operate in conventional vehicle routing as well. Finally, we aim to put the backpressure algorithm to a more complex road network, and perhaps to a real test.

REFERENCES

- Abousleiman, R. and Rawashdeh, O. (2014). Energy efficient routing for electric vehicles using particle swarm optimization. Technical report, SAE Technical Paper.
- Afroditi, A., Boile, M., Theofanis, S., Sdoukopoulos, E., and Margaritis, D. (2014). Electric vehicle routing problem with industry constraints: trends and insights for future research. *Transportation Research Proceedings*, 3:452–459.
- Artmeier, A., Haselmayr, J., Leucker, M., and Sachembacher, M. (2010). The optimal routing problem in the context of battery-powered electric vehicles. In *Workshop CROCS at CPAIOR-10, 2nd International Workshop on Constraint Reasoning and Optimization for Computational Sustainability*.
- Baheti, R. and Gill, H. (2011). Cyber-physical systems. *The impact of control technology*, 12:161–166.
- Baum, M., Dibbelt, J., Gemsa, A., and Wagner, D. (2014). Towards route planning algorithms for electric vehicles with realistic constraints. *Computer Science-Research and Development*, pages 1–5.
- Bondy, J. A. and Murty, U. S. R. (1976). *Graph theory with applications*, volume 290. Macmillan London.
- Bruglieri, M., Pezzella, F., Pisacane, O., and Suraci, S. (2015). A variable neighborhood search branching for the electric vehicle routing problem with time windows. *Electronic Notes in Discrete Mathematics*, 47:221–228.
- BUREAU, O. P. R. (1964). Traffic assignment manual. *US Department of Commerce*.
- Dantzig, G. B. and Ramser, J. H. (1959). The truck dispatching problem. *Management science*, 6(1):80–91.
- de Weerd, M. M., Stein, S., Gerding, E. H., Robu, V., and Jennings, N. R. (2015). Intention-aware routing of electric vehicles.
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271.
- Emadi, A. (2011). Transportation 2.0. *Power and Energy Magazine, IEEE*, 9(4):18–29.
- Jahn, O., Möhring, R. H., Schulz, A. S., and Stier-Moses, N. E. (2005). System-optimal routing of traffic flows with user constraints in networks with congestion. *Operations research*, 53(4):600–616.
- Kopetz, H. (2011). Internet of things. In *Real-time systems*, pages 307–323. Springer.
- Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416.
- Moeller, S., Sridharan, A., Krishnamachari, B., and Gnawali, O. (2010). Routing without routes: the backpressure collection protocol. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 279–290. ACM.
- Nagata, Y., Bräysy, O., and Dullaert, W. (2010). A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. *Computers & Operations Research*, 37(4):724–737.
- Neely, M. J., Modiano, E., and Li, C.-P. (2008). Fairness and optimal stochastic control for heterogeneous networks. *Networking, IEEE/ACM Transactions on*, 16(2):396–409.
- Neely, M. J. and Urgaonkar, R. (2008). Opportunism, backpressure, and stochastic optimization with the wireless broadcast advantage. In *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*, pages 2152–2158. IEEE.
- Patriksson, P. (1994). *The traffic assignment problem: models and methods*.
- Schneider, M., Stenger, A., and Goeke, D. (2014). The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, 48(4):500–520.
- Touati-Moungla, N. and Jost, V. (2012). Combinatorial optimization for electric vehicles management. *Journal of Energy and Power Engineering*, 6(5).

AUTHOR INDEX

Avgeriou, P.	3	Noutsos, A.	160
Bakelaar, R.	114	Oberhauser, R.	134
Bauer, B.	104	Orthacker, C.	196
Bender, B.	179	Papademetriou, R.	124
Colle, D.	213	Potzmader, K.	196
Cordeiro, J.	22	Pühringer, P.	196
Diamantopoulos, T.	160	Rauscher, J.	104
Fernández-Ledesma, J.	221	Roubtsova, E.	114
Futatsugi, K.	39	Shishkov, B.	187, 207
Gargouri, F.	84	Sikkel, N.	54
Glaschke, C.	179	Sinnhofer, A.	196
Greff, T.	64	Spruytte, J.	213
Gronau, N.	98, 179	Spyrou, E.	235
Gustas, R.	11	Staring, A.	54
Gustiené, P.	11	Stavrinides, G.	144
Hassen, M.	84	Steger, C.	196
Heikkilä, S.	227	Suurmond, C.	32
Henda, H.	217	Symeonidis, A.	160
Ibtissem, F.	217	Thim, C.	98
Janssen, M.	187	Tsagakani, C.	173
Johann, D.	64	Tsalgatidou, A.	173
Jürjens, J.	5	Turki, M.	84
Karatza, H.	144	Ullrich, A.	98
Karras, D.	124	Van der Wee, M.	213
Kirchmer, M.	7	Vartiainen, A.	227
Kommeri, J.	227	Verbrugge, S.	213
Komoto, T.	39	Vladova, G.	98
Kreiner, C.	196	Weber, E.	98
Langermeier, M.	104	Wedemeijer, L.	46
Maciaszek, L.	152	Werth, D.	64
Meertens, L.	54	Yemna, S.	217
Michell, V.	73	Yoshioka, N.	39
Mitrakos, D.	207, 235		
Niemi, T.	227		



Proceedings of BMSD 2016
Sixth International Symposium on Business Modeling and Software Design
ISBN: 978-989-758-190-8
<http://www.is-bmsd.org>