BMSD 2014

Fourth International Symposium on

Business Modeling and Software Design

Proceedings

Luxembourg, Grand Duchy of Luxembourg • 24-26 June 2014

Organized by:

IICREST

In Collaboration with:

tudor

PUBLIC RESEARCH CENTRE HENRI TUDOR

Cooperating Organizations:

SIKS

ARISTOTLE UNIVERSITY OF THESSALONIKI

CTIT
Centre for Telematics and Information Technology

AMAKOTA

BMSD 2014

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

Luxembourg, Grand Duchy of Luxembourg

24-26 June 2014

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

In Collaboration with

TUDOR - Public Research Centre Henri Tudor

Cooperating Organizations:

SIKS - the Dutch Research School for Information and Knowledge Systems

AUTH - Aristotle University of Thessaloniki

CTIT - Center for Telematics and Information Technology

AMAKOTA Ltd.

Copyright © 2014 SCITEPRESS - Science and Technology Publications
All rights reserved

Edited by Boris Shishkov

Graphics Production by Bozhana Yankova

Compiled in Portugal

Printed in Bulgaria

ISBN: 978-989-758-032-1

Depósito Legal: 374971/14

http://www.is-bmsd.org

secretariat@iicrest.org

BRIEF CONTENTS

KEYNOTE SPEAKERS

Hend*erik* Proper

Public Research Centre Henri Tudor

Grand Duchy of Luxembourg

Roel Wieringa

University of Twente

The Netherlands

CHAIR AND PROGRAM COMMITTEE

CHAIR

Boris Shishkov, 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

Ognyan Andreev, Technical University of Sofia, Bulgaria

Paulo Anita, Delft University of Technology, The Netherlands

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

Joop De Jong, University of Applied Sciences Utrecht, The Netherlands

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

Teduh Dirgahayu, Universitas Islam Indonesia, Indonesia

Lyubka Doukovska, Bulgarian Academy of Sciences, Bulgaria

Chiara Francalanci, Politecnico di Milano, Italy

PROGRAM COMMITTEE (CONT.)

Boris Fritscher, University of Lausanne, Switzerland

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

Arash Golnam, EPFL, 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

Dmitry Kan, AlphaSense Inc., Russia

Dimitris Karagiannis, University of Vienna, Austria

Marite Kirikova, Riga Technical University, Latvia

Samuel Kounev, Karlsruhe Institute of Technology, Germany

José Paulo Leal, University of Porto, Portugal

Kecheng Liu, University of Reading, UK

Leszek Maciaszek, Wroclaw University of Economics, Poland

Jelena Marincic, University of Twente, The Netherlands

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

Dimitris Mitrakos, Aristotle University of Thessaloniki, Greece

Preslav Nakov, Qatar Computing Research Institute - Qatar Foundation, Qatar

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

Sietse Overbeek, University of Duisburg-Essen, Germany

Mike Papazoglou, Tilburg University, The Netherlands

Marcin Paprzycki, Polish Academy of Sciences, Poland

Oscar Pastor, Universidad Politécnica de Valencia, Spain

Hend*erik* Proper, Public Research Centre - Henri Tudor, Grand Duchy of Luxembourg

Ricardo Queirós, IPP, Portugal

Jolita Ralyte, University of Geneva, Switzerland

Gil Regev, EPFL / Itecor, Switzerland

PROGRAM COMMITTEE (CONT.)

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, Bilkent University, Turkey

Linda Terlouw, ICRIS B.V., The Netherlands

Yasar Tonta, Hacettepe University, Turkey

Roumiana Tsankova, Technical University of Sofia, Bulgaria

Marten van Sinderen, University of Twente, The Netherlands

Maria Virvou, University of Piraeus, Greece

Roel Wieringa, 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 ten selected papers presented at BMSD 2014 will be invited by Springer-Verlag to submit revised and extended versions of their papers for publication in a Springer LNBIP Series book.

FOREWORD

When developing an information system, we need adequate underlying business / 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. Even though some results have been delivered in this direction, we do not observe actual improvements yet and still many information systems development projects go over time/budget, and the user satisfaction remains low. Moreover, the greater current computing and hardware possibilities inspire developers to offer more and more "built in" intelligence for the benefit of the user, which is nevertheless causing even lower user satisfaction and even frustration – the ways in which many software systems "think" for the user, "decide" what should be the needs of the user, "adapt" to inferred user habits, and so on, are widely unacceptable for the public. More and more we hear people saying that human behavior is too complex to be graspable by a programmed software system in an adequate and nearly exhaustive way. At the same time, we believe that we should not deny the inspiring progress software development has reached. Instead, we may better learn to what extent we should rely on software and we should improve our ability to align software to its real-life (business) context – here enterprise models are claimed to be crucial (not only for understanding and/or (re-)engineering the enterprise under consideration but also for adequately developing on top a supportive software system). Automating (partially) enterprise processes by means of software systems can only be accomplished in a methodological and systematic way, if based on corresponding enterprise models. Said otherwise, software generation should stem from corresponding enterprise modeling, by aligning enterprise modeling concepts and corresponding software specification concepts. Only such an enterprise-software alignment could actually guarantee that: (a) the software system would be properly integrated in its enterprise context; (b) an enterprise-software traceability would be possible allowing not only for software updates driven by new requirements but also for possible enterprise re-engineering activities, inspired by a goal to better fit the enterprise to the IT platform(s) used; (c) re-use could be an issue, counting on enterprise modeling constructs and software components. This points to the emerging discipline of enterprise engineering that addresses challenges such as agility and adaptability of enterprises, and is partially inspired by the goal of better facilitating the development of supportive (with regard to enterprises) software systems. Nevertheless, most current enterprises represent complex, evolving networked organizations while most current software systems are composed of sophisticated components and driven by complex rules. Thus, neither enterprise models nor software specification models can usefully be built starting from the scratch – what we need are (generic) enterprise modeling patterns and possibility to adequately reflect such patterns in corresponding software components. Closing the gap between enterprise modeling and software design in a component-based way has been an important research challenge for years already. All this has been dominant for BMSD, the international symposium on Business Modeling and Software Design, bringing together researchers and practitioners interested in business/enterprise modeling and its relationship to software design, and demonstrating for a fourth consecutive year a high quality of papers and presentations as well as a stimulating discussion environment. The theme of BMSD 2014 (the Fourth International Symposium on Business Modeling and Software Design) is: "GENERIC BUSINESS MODELING PATTERNS AND SOFTWARE RE-USE", and the scientific areas of interest to the symposium are: (a) business models and requirements; (b) business models and services; (c) business models and software; (d) information systems architectures. Further, there are three application-oriented special sessions, namely: a special session on e-Health Services and Technologies, a special session on Intelligent Systems and Business Analysis, and an Industrial Track. These special sessions are bringing additional practice-driven value to the symposium.

This book contains the proceedings of BMSD 2014, held in Luxembourg, Grand Duchy of Luxembourg, on 24-26 June 2014. The proceedings consists of 37 high-quality research and experience papers that have not been published previously. These papers have undergone a detailed peer-review process and were selected based on rigorous quality standards.

The symposium has been organized and sponsored by the Interdisciplinary Institute for Collaboration and Research on Enterprise Systems and Technology (IICREST), in collaboration with the Public Research Centre Henri Tudor (TUDOR). Cooperating organizations have been the Dutch Research School for Information and Knowledge Systems (SIKS), Aristotle University of Thessaloniki (AUTH), the UTwente Center for Telematics and Information Technology (CTIT), and AMAKOTA Ltd.

The fourth edition of BMSD follows three inspiring events, namely: Sofia 2011, Geneva 2012, and Noordwijkerhout 2013. We are proud to have succeeded in establishing and maintaining high scientific quality and stimulating collaborative atmosphere – the BMSD Community is characterized by competence, motivation, sharing, and innovativeness. In addressing the above-mentioned research challenges and areas, BMSD 2014 has considered a large number of research topics: from more conceptual ones, such as enterprise modeling, modeling languages, meta-modeling, ontologies, business rules, enterprise regulations, model-driven adaptability, intelligent systems, and Semiotics, to more technical ones, such as software specification, use cases, database clusters, model-driven testing, and 'e-applications' (in Healthcare and Business), from more business-oriented ones, such as business model design, enterprise architecture management, business process simulation, enterprise resource planning and strategies, and requirements specification, to software architectures –related topics. We believe that all these research contributions highlight challenging (technical) problems and present innovative solutions relevant to the scientific areas mentioned already.

The 37 published papers (including several Invited Papers) were selected from 52 submissions and 12 of these papers were selected for a 30-minutes oral presentation (Full Papers); in addition, 25 papers were selected for a 20-minutes oral presentation (Short Papers and Special Sessions Papers).

Hence, the full-paper acceptance ratio of 23% (the same as in 2013) shows a high level of quality which we intend to maintain and reinforce in the following editions of the symposium. Further, the BMSD'14 authors are from: Austria, Belgium, Bulgaria, China, Germany, India, Ireland, Italy, Japan, Kazakhstan, Luxembourg, The Netherlands, Poland, Portugal, Russia, Spain, Switzerland, Taiwan, Tunisia, UK, and USA (listed alphabetically); this makes in total 21 countries (compared to 14 countries having been represented in 2013, 11 countries – in 2012, and 10 countries – in 2011); 7 countries, nevertheless, have been represented in all 4 BMSD editions so far, these are: Belgium, Bulgaria, Germany, The Netherlands, Switzerland, Russia, and UK. This clearly indicates for a very strong European influence and also for our succeeding to add on the "BMSD Map" impressive non-European countries.

The current proceedings' Publisher is SCITEPRESS and we deliver not only printed proceedings but also an electronic version of the proceedings – all presented papers will be made available at the SCITEPRESS Digital Library by September, 2014. Furthermore, the proceedings will be submitted for indexation by DBLP (Computer Science Bibliography). Finally, the authors of around ten selected papers presented at BMSD 2014 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 2014 program is enhanced by two Keynote Lectures, delivered by distinguished guests who are renowned experts in their fields, including (alphabetically): Hend*erik* Proper (Public Research Centre Henri Tudor, Grand Duchy of Luxembourg) and Roel Wieringa (University of Twente, The Netherlands). In addition, the Keynote Speakers and other BMSD'14 participants will take part in a panel discussion and also in other discussions stimulating community building and facilitating possible R&D project acquisition initiatives. These high points in the symposium program would definitely contribute to maintaining the event's high quality and its stable and motivated Community.

Building and interesting and successful program for the symposium required the dedicated efforts of many people. Firstly, we must thank the Authors, whose research and development achievements are recorded here. Also, 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 necessary work for delivering a stimulating and productive event; the greatly appreciated support of Dimitris Mitrakos is to be especially mentioned here. We appreciate the local support brought forward by our Colleagues from the Public Research Centre Henri Tudor. We appreciate as well the willingness of SCITEPRESS to publish the current proceedings and we bring forward special compliments to Vitor Pedrosa for his devoted and professional work with regard to the proceedings preparation. We are indebted to Alexander Verbraeck from TU Delft for all his encouraging support. 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 the beautiful city of Luxembourg. We look forward to seeing you next year in Milan, Italy, for the Fifth International Symposium on Business Modeling and Software Design (BMSD 2015), details of which will be made available at http://www.is-bmsd.org.

Boris Shishkov
IICREST, Bulgaria

CONTENTS

KEYNOTE SPEAKERS

FULL PAPERS

SHORT PAPERS

SPECIAL SESSION ON E-HEALTH SERVICES AND TECHNOLOGIES (EHST)

SPECIAL SESSION ON INTELLIGENT SYSTEMS AND BUSINESS ANALYSIS

INDUSTRIAL TRACK

KEYNOTE SPEAKERS

Exploring the Challenges of Modelling Landscapes

Hend*erik* Proper
Public Research Centre Henri Tudor, Grand Duchy of Luxembourg
erik.proper@tudor.lu

Abstract: In enterprise modelling, a wide range of models and languages is used to support different purposes. If left uncontrolled, this can easily result in a fragmented perspective on the enterprise, its processes and IT support. On its turn, this negatively affects traceability, the ability to do cross-cutting analysis, and the overall coherence of models. Different strategies are suggested to achieve model integration. They mainly address syntactic-semantics aspects of models/languages, and only to a limited extent their pragmatics. In actual use, the 'standardising' and 'integrating' effects of traditional approaches (e.g. UML, ArchiMate) erodes. This is typically manifested by the emergence of local 'dialects', 'light weight versions', as well as extensions of the standard to cover 'missing aspects'. This presentation aims to create more awareness of the factors that are at play when creating integrated modelling landscapes. Relying on our ongoing research, we develop a fundamental understanding of the driving forces and challenges related to modelling and linguistic variety within modelling landscapes. In particular, the presentation discusses the effect of *a priori* fixed languages in modelling and model integration efforts, and argues that they bring about the risk of neglecting the pragmatic richness needed across practical modelling situations.

BRIEF BIOGRAPHY

Prof.dr. Henderik A. Proper, Erik for friends, is a senior research manager at Public Research Centre - Henri Tudor in Luxembourg. He is also professor of information systems at the Radboud University Nijmegen, The Netherlands. He is one of the co-initiators of the development of the ArchiMate language for Enterprise Architecture. At Tudor, he leads the research programme on enterprise engineering. Erik has co-authored two books on enterprise architecture, and provided substantial contributions to two other books on this topic. He is also an editor in-chief of the book series on Enterprise Engineering, published by Springer. His home on the web can be found at "http://www.erikproper.eu".

The Structure of Goal Models in Requirements Engineering

Roel Wieringa
University of Twente, The Netherlands
R.J.Wieringa@utwente.nl

Abstract: Requirements engineering is the activity to mutually align business goals with software systems behavior. In goal-oriented RE, goal models are used in a systematic process to exploit the interaction between the properties of the business context of a software system and properties of the system itself in order to contribute to stakeholder goals. In this talk I will describe this process and give examples to illustrate it. I then highlight several aspects of the structure of goal models, including means-end structures, temporal structures and contribution structures, and show what role they play in reasoning about organization design and software design.

BRIEF BIOGRAPHY

Roel Wieringa (http://www.cs.utwente.nl/~roelw) is Chair of Information Systems at the University of Twente, The Netherlands. His research interests include requirements engineering, risk assessment, and design research methodology. He has written two books, on Requirements Engineering and on the Design of Reactive Systems. His next book, Design Science Methodology for Information Systems and Software Engineering will appear in 2014 with Springer.

FULL PAPERS

Power-Modelling
Toward a More Versatile Approach to Creating and Using Conceptual Models

Ulrich Frank
University of Duisburg-Essen, Essen, Germany
ulrich.frank@uni-due.de

Keywords: DSML, Interactive Modelling, Multi-level Modelling, Empowerment.

Abstract: The prospects of conceptual modelling are widely undisputed. Nevertheless the current practice of conceptual modelling remains unsatisfactory. Usually, modelling languages offer primitive concepts only—with respective effects on productivity and model quality. The creation of models is restricted to early phases of system life-cycle. Hence, the benefits of models in later phases are ignored. Furthermore, the creation and use of conceptual models is still restricted to experts only. In this paper, the outline of a new modelling paradigm, referred to as power-modelling, is presented. It builds on the potential of domain-specific modelling languages (DSML), application frameworks and reference models. It regards models as the primary medium to perceive, interact with and change systems and the environment they are supposed to operate in during the entire system life-cycle. For this purpose, power-modelling is built on an extensible set of multi-level DSML that fit the conceptual perspectives of a wide range of prospective users and a common representation of models and code, which allows overcoming the notorious problem of synchronizing models and code.

1 INTRODUCTION

If maturity comes with age, a few decades of research and application should have resulted in a widely perfected state of conceptual modelling. There are indeed various signs of maturity. Both in Information Systems and Computer Science it seems undisputed that conceptual models are a prerequisite to manage the complexity of large software systems. Furthermore, it is acknowledged that conceptual models are much better suited than code to involve prospective users and other stakeholders in the process of developing software. The benefits of conceptual models for designing software systems do not come as a surprise. After all, software systems are *linguistic artefacts*. On the one hand, they are realized with some kind of implementation language. On the other hand, as non-physical artefacts they can be perceived—and used—by humans only through some kind of linguistic representation. At best, this linguistic representation corresponds to the language used in the targeted domain. Conceptual models are aimed at reconstructing domain-specific languages in a way that prospective users perceive them as familiar, while at the same time they allow for transformations into implementation-level languages. By structuring and eventually automating the transformation of models into code, as it is pursued by approaches to model-driven software development (Atkinson and Kühne, 2003; France and Rumpe, 2007), conceptual modelling is promising to substantially improve the productivity of developing software systems.

But conceptual modelling is not restricted to modelling software systems. Exploiting the potential of information systems often requires re-organizing respective action systems. Consequently, corresponding modelling approaches, such as business process modelling, turned out to be a good choice for supporting people with analysing and (re-) structuring action systems. The insight that corporate action systems do not only comprise business processes, but also other subjects, such as goals, resources or organizational structure, contributed to the emergence of approaches to enterprise modelling (Scheer, 1992; Ferstl and Sinz, 2006; Frank, 2013). By integrating conceptual models of software systems with conceptual models of action systems, enterprise models promise to provide a foundation for jointly analysing and designing information systems and the relevant organizational context. Since more and more organizations have given up developing software on their own, the original idea of using conceptual models for designing software systems does not fit the demands of many organizations anymore. At the same time, the ever growing complexity of IT infrastructures created additional challenges. Again, conceptual models of IT infrastructures, including various representations of their high-level structure or architecture, have evolved as a remedy. While respective models are often part of enterprise modelling methods, they are also featured by approaches to enterprise architecture, which in general aim mainly at managers and therefore emphasize a higher level of abstraction (Lankhorst, 2005; Proper et al., 2010; Buckl et al., 2010). Therefore, conceptual modelling supports a wide range of activities related to analysing, designing and managing IT infrastructures and corresponding action systems. A relatively large research community, both in Information Systems and Computer Science, may serve as further evidence for the maturity of the field.

However, our brief assessment of the field's contributions may be deceptive—and maturity may have its downsides, too. As we shall see, there are serious reasons for not being satisfied with the state of the art in conceptual modelling. There are even reasons to challenge the current paradigm of conceptual modelling. In any case, it does not seem appropriate to further follow existing paths of research by focussing solely on problems within this paradigm without leaning back once in a while to reflect upon other ways to conceptualize, develop, maintain and use models. While I had my doubts for some time whether we are doing the right thing, I started to intentionally abandon some characteristics of conceptual modelling which I had not only taken for granted, but which I used to preach enthusiastically. I am still in the—sometimes painful, sometimes exciting—process of realigning my perspective and my research agenda. However, a few ideas have emerged that I regard as promising. One of them is subject of this paper. I dared calling it "power-modelling", not only to express that it might be suited to substantially promote the power of modelling as a tool to create and modify systems, but also to empower users by supporting them with sophisticated, but convenient instruments to use and modify the systems they work with.

The paper starts with a critical review of the current state of the art. Subsequently, an overview of approaches that address certain shortcomings of conceptual modelling is given. Against this background, I will describe the idea of power-modelling by outlining the foundational concepts and by illustrating how it could be implemented. Since the respective research is in a very early stage and faces serious challenges, the conclusions will especially focus on future research.

2 CONCEPTUAL MODELLING: A CRITICAL REVIEW

There are various reasons, why the current state of conceptual modelling might not be regarded as satisfactory (Frank, 2014a). At first, there is the sobering fact that the dissemination of conceptual modelling in practice is still rather modest. Many software developers still regard it as dispensable. Managers perceive it as a cost-driver that does not deliver measurable benefit. Finally, prospective users are often not keen to look into conceptual models, nor are they capable of designing them on their own. The unsatisfactory adoption in practice may be contributed to a lack of respective professional education among today's workforce. However, I am afraid, we would take the easy way out, if we were satisfied with this explanation. There are other, more essential reasons for questioning the power of current approaches to conceptual modelling, which also may, in part, hinder its acceptance and dissemination in practice. They relate to the economics of modelling and the psychological assumptions underlying the construction and perception of conceptual models.

2.1 Economics of Modelling

With respect to economics of modelling, three aspects are of particular relevance. At first, there is the productivity of creating, analysing and modifying conceptual models, i.e. the time these activities take for a certain outcome. While modelling productivity depends on modellers' skills and experiences, it is also affected by the available modelling languages and tools. A modelling language can contribute to productivity by providing reusable artefacts and by allowing for abstractions that foster reuse and adaptation of models. Currently, most modelling languages are restricted to a few semantic primitives that remind of basic ontologies like the one suggested by (Bunge, 1977) or (Grossmann, 1983). While generic concepts such as "entity type", "class", "attribute", etc. can be applied to any domain—which is the purpose of a general ontology—they require modellers to reconstruct all concepts of a model from scratch. Hence, they promote a wide range of reuse, which should improve economies of scale e.g. of modelling tools, but in a particular case, their contribution to productivity is very poor. Imagine, you would have to describe a domain such as accounting and you were restricted to a language that consisted of generic concepts like the ones provided by the ERM and the UML! Furthermore, modelling languages provide only a few abstraction concepts such as classification, generalization or encapsulation. As a consequence, similarities between a range of models can often not be accounted for by abstracting on a set of common properties. The lack of abstraction is especially painful in business process modelling, where reuse is widely restricted to copy&paste (Frank, 2012). Second, there is little protection of investment. The use of conceptual models is widely restricted to the build time phase. In later phases, models are used for documentation only, if they are used at all. Even in those cases, where models were used to generate code, they will usually get devalued over time, because during maintenance changes are directly applied to code and synchronizing models and code, beside representing a serious challenge, does not happen. Third, the benefit of models depends on their quality. However, judging the quality of conceptual models is a demanding task and requires experts. Current modelling languages and tools hardly contribute to model quality, since their generic concepts allow for almost any kind of absurd models as long as they are syntactically correct. From a managerial perspective there is the additional problem that the economics of modelling is hard to judge. While it is often not trivial to determine modelling costs in advance, quantifying the benefits of models ex ante is almost impossible. As long as managers are not convinced that modelling is suited to generate substantial benefit, the lack of legitimate quantification methods creates a serious obstacle to conceptual modelling in practice.

2.2 Cognitive Capabilities

Conceptual modelling is based on basic assumptions that most members of the modelling community—including myself—are convinced of. First, there is the assumption that the analysis and design of complex systems recommends a rational approach, i.e. an approach that is characterized by the differentiated consideration of pros and cons and that puts emphasis on justifying decisions. Second, related to the first assumption, following a Kantian tradition, a rational perspective demands for focussing on *concepts*. Conceptual modelling is typically aimed at the *reconstruction* of terms used in the domain of interest. Referring to existing terminology is supposed to make conceptual models accessible by people working in the domain. At the same time, a reconstruction is required in order to create concepts that fit the specific modelling purpose and that facilitate mapping to implementation languages. As a consequence, it it common to specify concepts according to the definitions of types or classes in implementation languages. While these basic ideas make perfect sense, they are based on an idealization that is seriously challenged by research in cognitive psychology. There is a plethora of work that shows the rather limited ability of most humans to precise and consistent thinking, for an overview see (Kahneman et al., 1982). Furthermore, the way people acquire and use concepts is often in clear contrast with the definitions we use in conceptual modelling (Lakoff, 1990). Among the most relevant insights are the following: Concepts are often not associated with (extensional) definitions, but rather with a few typical examples (prototypes"). Conceptual categories are often perceived as having no clear boundaries (membership gradience", (Lakoff, 1990), p. 12). The last example is of especial relevance for our investigation: Most cognitive models are embodied with respect to use." ((Lakoff, 1990), p. 12) As a consequence, we cannot expect prospective users to be much interested in and capable of thinking in concepts as we use them for modelling purposes.

2.3 Preliminary Conclusion

Our brief analysis of the current state of conceptual modelling resulted in a number of problems. That does not mean, however, to question the idea of conceptual modelling in general. In order to cope with complexity and change there is no alternative for us other than to somehow develop models. Without abstraction not only our understanding of systems and of the world in general will remain poor, but so would be our ability to design and implement systems. Nevertheless, those problems should make us think. In particular, there are three areas that demand for more attention. There is need to promote modelling productivity. For this purpose, reuse has to be fostered by providing modelling constructs that incorporate domain-specific semantics. To promote model quality, the reusable artefacts offered to users should be thoroughly developed and tested. At the same time, economies of scale are crucial. For this purpose, an artefact needs to be reused in a wide range of cases, which may create a conflict to modelling productivity. To improve the involvement of prospective users, models, new forms of presenting concepts and models to users are required. Since there is evidence that people imagine or learn concepts by associating them with contexts of use, it might be advisable to develop modelling environments that focus on the use of concepts. The use of models during later phases of the system life-cycle demands for developing scenarios that illustrate the respective benefit. In addition, technical challenges, such as solving the notorious problem of synchronizing code and model have to be addressed.

3 SELECTED APPROACHES

There have been various approaches that address some of the problems elucidated above. Some of them are aimed at improving modelling productivity and model quality, while others focus on implementation issues or on user involvement.

3.1 Focus on Productivity and Quality

Reference models are based on a convincing idea. By developing models that serve as a reference for certain domains, the costs for realizing large models in respective domains could be reduced substantially. At the same time, reference models should be developed with outstanding care and expertise. Therefore, they should effectively improve model quality. Furthermore, reference models do not only stress a descriptive intention, i.e. representing a domain as it is, but also a prescriptive intention, i.e. representing conceptualisations that seem especially favourable for the future development of organizations in the targeted domain. Despite their convincing foundation, the dissemination of reference models in practice remained extremely modest. Domain-specific modelling languages (DSMLs) are build on a similar idea. Different from general-purpose modelling languages (GPMLs), they comprise domain-specific concepts, thus freeing modellers from the need to specify these concepts from scratch. At the same time, DSMLs feature usually, but not necessarily, a concrete syntax that is adapted to representations known in the domain they cover. In addition to promoting productivity through improved reuse and ergonomics, DSML also contribute to model quality, since their syntax and semantics facilitate preventing models that are all too strange. Modelling is more convenient, and the range of possible models is clearly restricted compared to a GPML.

While DSMLs are certainly suited to address the productivity challenge, they are not the silver bullet of conceptual modelling, since they face a fundamental conflict of system design. On the one hand, increasing productivity demands for concepts that are specific to a particular domain, i.e. that incorporate a high degree of domain-specific semantics. On the other hand, economies of scale demand for DSMLs that cover a wider domain, i.e. that include a smaller degree of domain-specific semantics. This fundamental conflict of designing DSMLs is illustrated in fig. 1

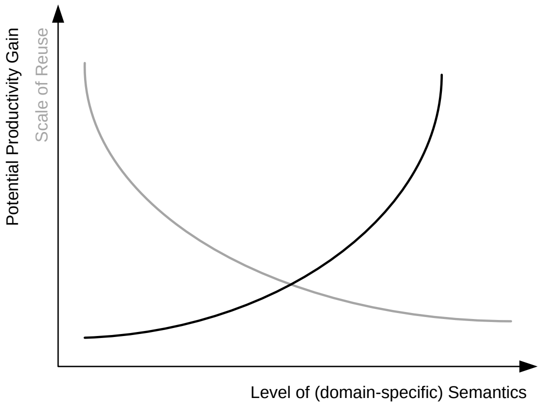


Figure 1: The fundamental conflict of designing DSMLs.

3.2 Focus on Implementation

If one regards a model mainly as an instrument for developing software, the transformation of models into code is a major aspect of increasing the value of models. Some time ago, Wiederhold et al. proposed an approach, called “megaprogramming” to substantially promote the productivity of *programming* large application systems (Wiederhold et al., 1992). Even though they do not explicitly speak of modelling, there are clear relations to conceptual modelling in their work. They suggest to compose large software systems from “megamodules”. Megamodules can be thought of as domain-specific abstraction: megamodules “capture the functionality of services provided by large organizations like banks, airline reservation systems, and city transportation systems. ... The concepts, terminology, and interpretation paradigm of a megamodule is called its ontology.” (Wiederhold et al., 1992), p. 89. However, the representation of megamodules as well as their composition happens on the code level only. Therefore, they are hardly suited for being used by non-experts. It is, however, conceivable to combine the idea of megamodules with a representation that corresponds more clearly to a terminology users are familiar with. Even though megamodules, like DSML, promote domain-specific artefacts, their use is based on a bottom-up approach in the sense that a system is created from some kind of building block without providing a blueprint for the overall design. *Application frameworks* stress a top-down approach. An application framework represents an architecture and the partial implementation of a class of application systems. “Black box” frameworks can be adapted only through interfaces, while “white box” frameworks allow for modifying their code. Frameworks can substantially boost the productivity of application development. However, while the adaptability of black box frameworks is limited, using white box frameworks efficiently requires considerable effort. Similar to megamodules the representation of frameworks is usually restricted to code. Approaches to *model-driven software development* take advantage of conceptual models for designing systems. At the same time, they promote implementation productivity by aiming at generating software systems from models (France and Rumpe, 2007), (Stahl and Völter, 2006). To cope with a multitude of platforms and programming languages, specific effort has been put on generating platform- and language-independent representations that allow for a straightforward transformation into particular implementations (Mellor, 2004), (Pastor and Molina, 2007). Even though model-driven software development is suited to improve development productivity and software quality, it remains unsatisfactory with respect to the evolution of software systems: Usually it is not possible to generate an entire software system from models, i.e. there is need for manual extensions. As a consequence, the evolution of code and models has to be synchronized. While there are a few approaches that address the challenge of synchronizing models and code (e.g. (Balz et al., 2010), (Agrawal, 2003)), they are not satisfactory, because they cannot always ensure consistency.

3.3 Focus on User Involvement

To some extent, the development of DSML and corresponding editors is aimed at making modelling more convenient for users who are not trained in conceptual modelling. The idea of using models at run time (Blair et al., 2009), while also contributing to the protection of investments, may be suited to motivate more people to use models: as a conceptual representation of the systems they interact with and may want to modify. However, recent research in this area is mainly focused on software engineering aspects such as synchronisation of models and systems, e.g. (Song et al., 2011), or self-adapting systems, e.g. (Amoui et al., 2012), (Morin et al., 2009). Krogstie proposes using models during the entire life-cycle of a system to emphasize user *empowerment* (Krogstie, 2007). For this purpose, modelling should not longer be restricted to system development. Instead, for modelling to have a "larger effect", he proposes "to enable all knowledge workers to be active modelers." (p. 305). Enterprise software systems should be presented to their users as "interactive models", the use of which is "about discovering, externalizing, capturing, expressing, representing, sharing and managing enterprise knowledge." (p. 306). In addition to known approaches to enterprise modelling, Krogstie stresses the need for more intuitive representations of models, such as "visual scenes for pro-action learning" and descriptions of the relevant context that focus on actions (p. 308). Using models as objects and objectivation of organizational knowledge work and of individual learning is appealing. However, Krogstie remains vague about the realization of his vision. He suggests a "model-generated workplace (MGWP)" that "is a working environment for the business users involved in running the business operations" (p. 312), but he does not provide details of how to accomplish "integrated modelling and execution platforms" (p. 308).

4 OUTLINE OF POWER-MODELLING

The idea presented in this section was inspired by various streams of work, some of which are described above. It is also a result of our long-standing-work on enterprise modelling that confronted us with some serious problem we were not able to solve as long as we were still bound to the traditional principles of conceptual modelling and of implementing modelling tools. Hence, the outline of power-modelling is also an attempt to suggest a new paradigm of creating and using conceptual models.

4.1 Objectives and Challenges

Against the background of our previous discussion, the following objectives mark desirable features of a future conception and realization of modelling.

Objective O1: A powerful approach to conceptual modelling should enable the use of models during the entire life-cycle of a system. *Rationale*: The complexity of enterprise systems demand for a representation that users are able to understand, not only to obtain a better comprehension of a software system, but also of the the context, since an increasing part of an enterprise is represented by its software systems. At the same time, people need (cognitive) models anyway to make sense of software systems and of organizations. Therefore, explicit models that fit the cognitive capabilities of users should be suited to increase organizational transparency. *Objective O2*: Conceptual models that represent software systems and the context they operate in should be *interactive*. *Rationale*: In order to serve as a universal interface to enterprise systems that fosters user empowerment, models need to allow for navigation, for searching and for modifying their states—not only during build time, but during run time. *Objective O3*: Conceptual models should provide access to their conceptual foundation, i.e. to the modelling language they are defined with, i.e. respective modelling tools should be *self-reflective*. *Rationale*: Users may want to get a better understanding of the concepts the models they interact with are based on. Furthermore, advanced users may even want to change those concepts. *Objective O4*: Conceptual models should be constructed from concepts that are used in professional discourse in the relevant domain. *Rationale*: Domain-specific concepts promote modelling productivity and make the use of modelling concepts more intuitive. *Objective O5*: Conceptual models of an enterprise should cover multiple perspectives and foster their integration. *Rationale*: The complexity of many organizations goes along with specialization which in turn results in different professional perspectives and languages. In order to satisfy the demand for providing users with representations they are familiar with, models of an enterprise need to account for different perspectives. On the one hand, that relates to providing modelling concepts which correspond to the technical language that is characteristic for a certain perspective. On the other hand, it should also be possible to present a model using different notations, both graphical and textual, in order to satisfy different cognitive styles. *Objective O6*: To increase the economics of modelling economies of scale have to be increased by promoting reuse. *Rationale*: Today, the effort it takes to develop elaborate models is often still prohibitively high. The above objectives create considerable challenges. Among those, three are hard to overcome within the current paradigm. Aiming at both, modelling concepts that reflect particular, domain-specific technical languages, and that promote economies of scale faces an obvious conflict: A language can either be more domain-oriented or built for serving more general purpose. Using models at run time that allow interacting with and eventually changing a software system demands a tight integration of models and code in order to keep their changes in synch. However, this is almost impossible due to limitations of prevalent programming languages. The elements of a model on M_1 are represented as objects on M_0, even though they are conceptually located on M_1. This is the case, too, for the representation of metaclasses in meta model editors. As a consequence, there is need to generate code (objects on M_0 cannot be further instantiated), which creates the notorious problem of synchronizing models and code. The lack of abstraction concepts in process modelling languages creates a serious obstacle for reuse. While generalisation/specialisation may be regarded as a suitable approach, it cannot be applied to processes in a straightforward way: To satisfy the substitutability constraint (Liskov and Wing, 1994), specialisation has to be monotonic, which is impossible to achieve for process models (Frank, 2012)

4.2 Cornerstones of Power-Modelling

For the vision of power-modelling to become real, the current paradigm is not sufficient. There is need to change our perspective on conceptual modelling and to aim at a different linguistic foundation—both of modelling and implementation languages. The following aspects mark cornerstones of a conception of power-modelling.

Emphasis on DSML: DSMLs that are reconstructed from existing technical terminologies (objective O4) promise clear advantages with respect to productivity, model quality and comprehensibility. For this purpose, we can build on an existing integrated set of DSMLs for enterprise modelling (Frank, 2013).

Multi-Level Modelling: In order to overcome the fundamental conflict of designing DSMLs (objectives O4 and O6), a *multi-level language* architecture is proposes. It is inspired by the definition and refinement of technical languages in practice. On a more generic, textbook" level, concepts are introduced that are supposed to fit a wide range of more specific domains. For that purpose, they remain intentionally abstract and underspecified. On more specific levels that could cover, for example, particular industries, concepts are refined and added. This process of stepwise refinement may in the end lead to numerous language levels, from more generic, over regional" to local" DSMLs. Multi-level modelling describes a language architecture that allows for an arbitrary number of classification levels. This allows to achieve both, a wide range of reuse, i.e. beneficial economies of scale, on higher levels, and a high productivity that is enabled by more specific DSML that reuse concepts of more generic DSML. Since all DSMLs are integrated in one language architectures, users can navigate all classification levels they are interested in (objective O3). Fig. 2 illustrates a multi-level language architecture and corresponding editors. For a detailed description of multi-level modelling see (Frank, 2014b).

Common representation of models and code: Using models during the entire life-cycle of a systems create the challenge of synchronizing models and code, which is caused by the fact that current implementation languages allow for one or two classification levels only. There are, however, a few languages, which are based on the recursive golden braid" architecture that facilitate systems with more classification levels. Among these languages, XMF (Clark et al.,

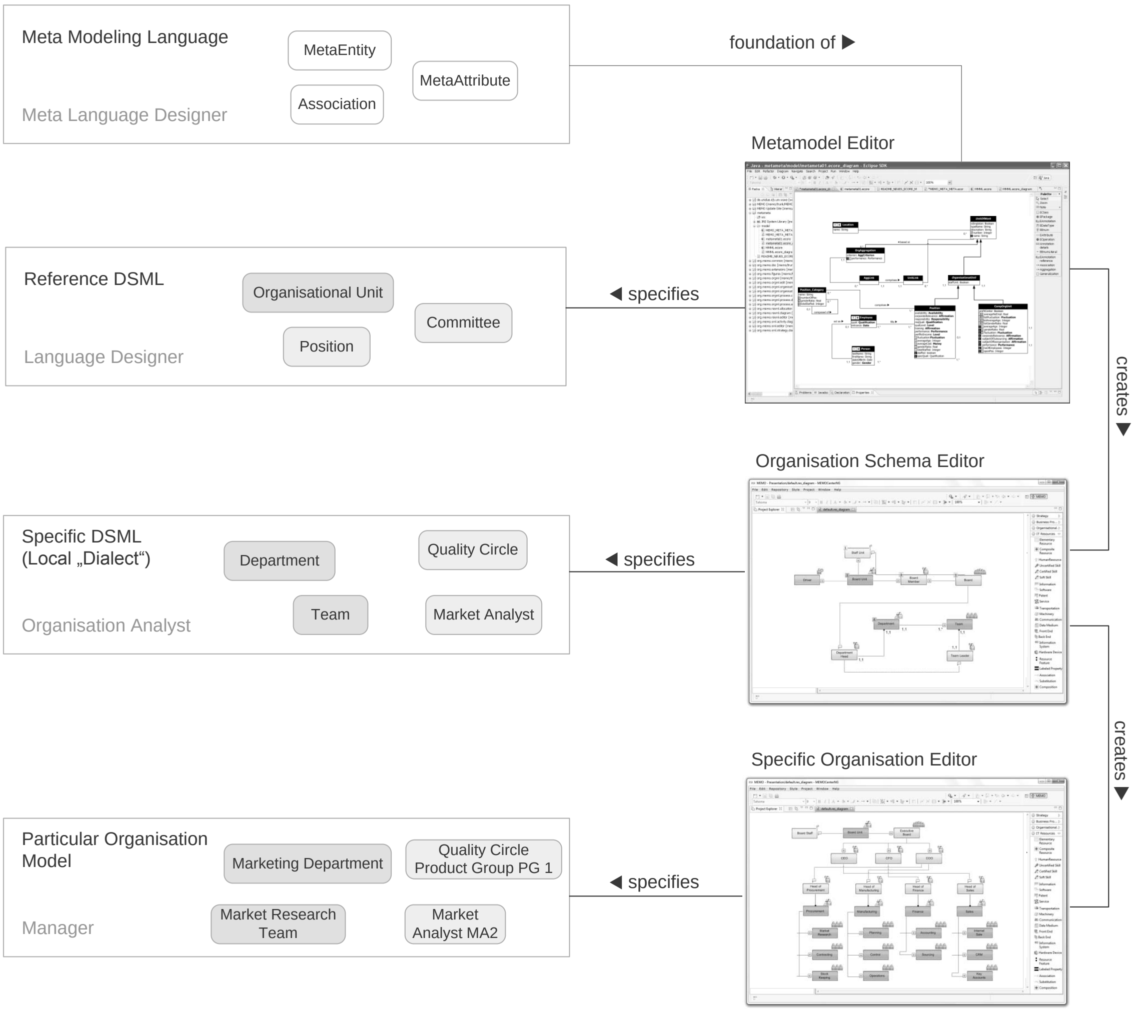


Figure 2: Illustration of multi-level language architecture.

2008b), (Clark et al., 2008a) is particularly suited for supporting power-modelling. First, it enables an arbitrary number of classification levels. Second, it is accompanied by a (meta) modelling environment, Xmodeler, that features a common representation of models and code, hence, it eliminates the need for synchronizing models and code and facilitates the use of models during the entire life-cycle of a system. In order to develop a suitable foundation of multi-level modelling and multi-level enterprise systems, we modified the metamodel of XMF (Frank, 2014b). By overcoming the separation of model and code, every system can be seen as a collection of models which can be used interactively (objective O2) with multiple representations that can be exchanged based on an implementation of the model-view-controller pattern (for details see (Clark et al., 2008a)).

Wider conception of conceptual modelling: In the existing paradigm, a conceptual model is created from the concepts of a modelling language, typically arranged in diagrams that represent some kind of a graph. However, modelling can also be thought of as an act of configuration that makes use of existing, more generic models and of DSML. In that case, modelling is more interactive, where a respective modelling system suggests alternative modelling options and requests user preferences. Since many users make sense out of action context rather than of singular concepts, interactive modelling should also go along with the representation of the context a model is supposed to address. A prototypical context could be provided by an enterprise model that integrates various perspectives on the enterprise (Frank, 2013) (objective O5). Furthermore, such a wider conception would include to not restrict modelling to drawing diagrams. Instead, models would be accessible through the elements of a graphical user interface, through tables or even through plain text. While it makes sense to re-

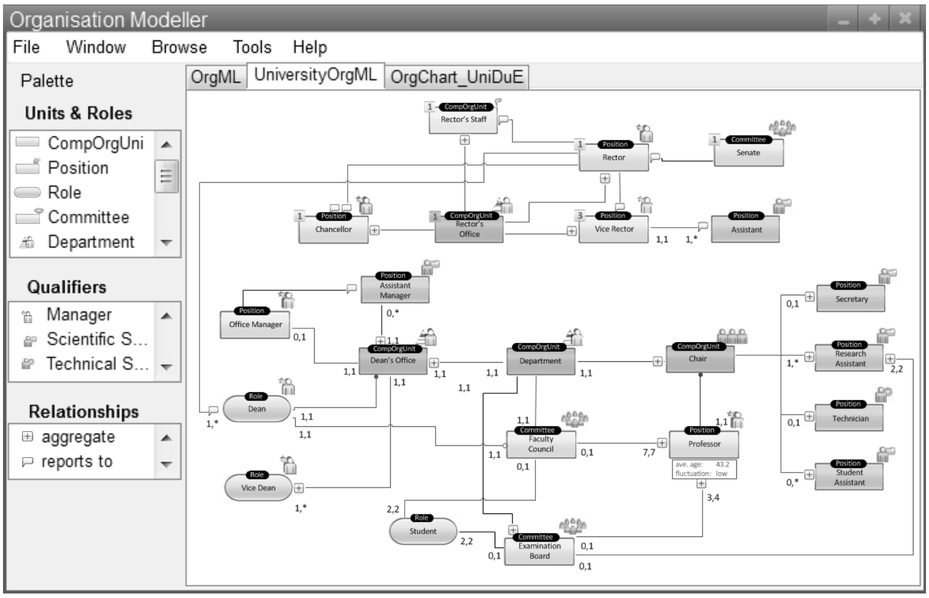


Figure 3: Definition of a DSML for modelling organizational structures of universities.

strict models to types or classes, a wider conception of modelling would also include models that comprise objects on M_0 using similar (graphical) representations to enable their interactive use. For example: The representation of a business process type could be supplemented by similar representations of business process instances (for an example in the context of so called "self-referential enterprise systems" see (Frank and Strecker, 2009), p. 14).

Ubiquitous use of (multi-perspective) models: Models are proposed as the primary medium to perceive, interact with and change systems—and the environment they are supposed to operate in during the entire system life-cycle. To serve this purpose, models have to be prepared for various perspectives to satisfy different skills and needs. Then they would no longer be a tool for system analysts and designers, but for everybody acting responsibly in an environment that is penetrated by information systems. As a consequence, power-modelling would contribute to empowering users by giving them (guided) access to a system's conceptual foundation and by supporting communication with other stakeholders.

4.3 Illustration

The following scenarios highlight selected aspects of power-modelling.

Refinement of DSML: Developing a DSML for modeling organizational structures may appear as a fairly trivial undertaking. However, this is not the case. First, a remarkable concept variance has to be accounted for. For example: A term like "department" may represent clearly different kinds if organizational units in different environments. In a university, a department consists of institutes and chairs. In some industrial enterprises, a department is part of a head department, in others not. Applying the idea of multi-level modelling to this domain would suggest developing a reference DSML for organization modelling. Such a reference DSML would include a descriptive graphical notation. A respective editor could be used by organization analysts to create an organizational schema for a certain domain, e.g. a particular organization or a range of organizations of a certain kind. Finally, this more specific DMSL would be used by local managers to build a particular organizational structure that corresponds to the previously defined schema. Fig. 3 illustrates the definition of a DSML for modelling organizational structures of universities by using a more general DSML that represents a textbook-level terminology.

The resulting DSML could then be used for creating an editor that facilitates modelling the organizational structure of a particular university. The resulting model is located on M_0. Even though it does not satisfy the abstraction usually demanded for by conceptual modelling, it is still beneficial to use a graphical notation that corresponds to that of the DSML defined on M_1. Such a model would enable interactive access to particular organizational units—and also allow for navigating to the specifications of respective concepts on higher classification levels.

Modelling as Interactive Configuration: Instead

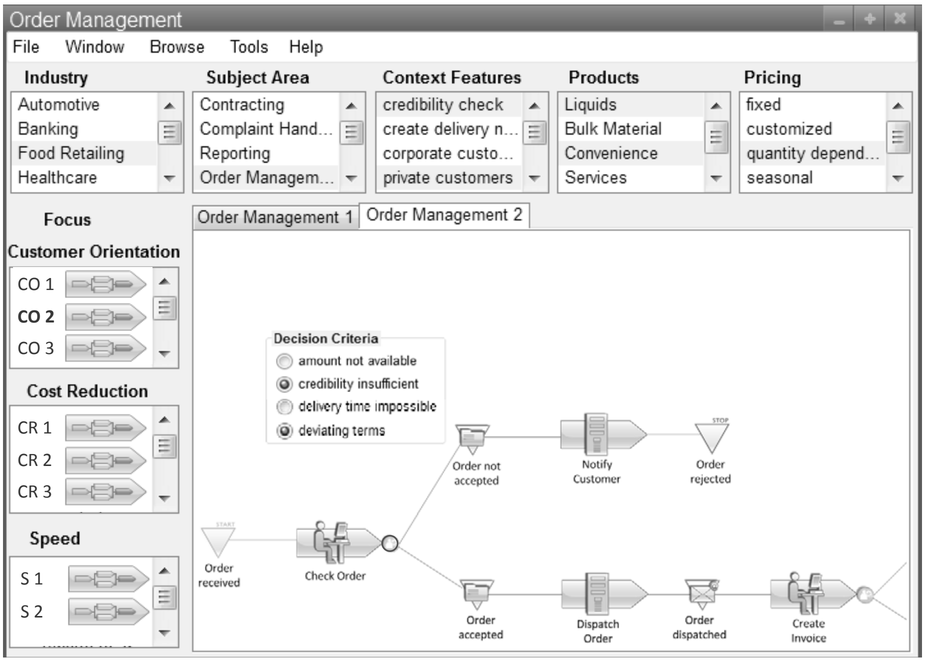


Figure 4: Configuration through stepwise selection of intended model properties.

of creating a model bottom-up from the concepts provided by a modelling language, the following scenario shows how a model can be created by combining reference models, application frameworks, multi-level modelling and a common representation of models and code to not only create models more conveniently but to realize a corresponding implementation at the same time. The scenario is based on the existence of a repository of various reference models that were created with a set of integrated DSMLs. It is aimed a modelling and thereby implementing an order management system for food retailing. Fig. 4 illustrates the use of a tool that facilitates the configuration of particular models from an existing set of reference models. At first, the user would specify the targeted domain by selecting options from lists presented by the modelling tool. Based on that selection, a set of possibly fitting business process types would be presented to the user. After selecting one business process type, the user could refine the process model by selecting from properties that are offered by the system. In case, the control flow requires further modification, the user could modify it using a respective DSML for business process modelling.

5 CONCLUSIONS AND FUTURE RESEARCH

The presented outline of a new approach to creating and using conceptual models has a twofold motivation. On the one hand, it stresses the need for leaning back once in a while and questioning assumptions that we tend to take for granted. On the other hand, it presents the cornerstones of a more versatile and powerful approach to conceptual modelling. The presentation of the vision is focussed on illustrating the idea in order to inspire a discussion about its benefits and about potential enhancements. Nevertheless, the required foundation, especially a multi-level language architecture and a common representation of models and code, is available. It has been developed during the recent years (Frank, 2014b), (Frank and Strecker, 2009) and builds on a mature meta-programming language (Clark et al., 2008b). While it required to give up certain assumptions that we had taken for granted, such as the rigid dichotomy of instantiation and specialization, it opens new ways of designing and implementing systems that are represented to users as models on different levels of abstraction. This paradigm shift requires not only giving up “standard” language architectures like MOF (Object Management Group, 2006), but also replacing existing implementation languages. Therefore, it may be regarded by some as not realistic. However, if we, at least in academia, do not give up the widespread fixation on mainstream technology, progress will hardly be possible. There is still a long way to go. Future research needs to address the development of reference models on different levels of abstraction that are specified with respective DSML. There is also need to develop advanced patterns of interaction that support the configuration and modification of models. Last but not least, it is required to overcome the limitations of current implementation languages by moving to more flexible languages architecture like the one XMF is based on.

REFERENCES

Agrawal, A. (2003). Metamodel based model transformation language to facilitate domain specific model driven architecture. In Crocker, R. and Steele, G. L. J., editors, *Companion of the 18th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA 2003)*, pages 118–119, New York. ACM.

Amoui, M., Derakhshanmanesh, M., Ebert, J., and Tahvildari, L. (2012). Achieving dynamic adaptation via management and interpretation of runtime models. *Journal of Systems and Software*, 85(12):2720–2737.

Atkinson, C. and Kühne, T. (2003). Model-driven development: a metamodeling foundation. *IEEE Software*, 20(5):36–41.

Balz, M., Striewe, M., and Goedicke, M. (2010). Continuous maintenance of multiple abstraction levels in program code. In *Proceedings of the 2nd International Workshop on Future Trends of Model-Driven Development (FTMDD 2010)*.

Blair, G., Bencomo, N., and France, R. B. (2009). Models@ run.time: Computer. *Computer*, 42(10):22–27.

Buckl, S., Matthes, F., Roth, S., Schulz, C., and Schweda, C. (2010). A conceptual framework for enterprise architecture design. In Proper, E., Lankhorst, M. M., Schönherr, M., Barjis, J., and Overbeek, S., editors, *Trends in Enterprise Architecture Research*, volume 70 of *Lecture Notes in Business Information Processing*, pages 44–56. Springer, Berlin and Heidelberg and New York.

Bunge, M. (1977). *Treatise on Basic Philosophy: Volume 3: Ontology I: The Furniture of the World*. Reidel, Dordrecht.

Clark, T., Sammut, P., and Willans, J. (2008a). Applied metamodelling: a foundation for language driven development.

Clark, T., Sammut, P., and Willans, J. (2008b). *Superlanguages: developing languages and applications with XMF*. Ceteva.

Ferstl, O. K. and Sinz, E. J. (2006). Modeling of business systems using som. In Bernus, P., Mertins, K., and Schmidt, G., editors, *Handbook on Architectures of Information Systems*, International Handbooks on Information Systems, pages 347–367. Springer, Berlin and Heidelberg and New York.

France, R. B. and Rumpe, B. (2007). Model-driven development of complex software: A research roadmap. In Briand, L. C. and Wolf, A. L., editors, *Workshop on the Future of Software Engineering (FOSE '07)*, pages 37–54. IEEE CS Press.

Frank, U. (2012). Specialisation in business process modelling: Motivation, approaches and limitations. icb research report, no. 51, university of duisburg-essen.

Frank, U. (2013). Multi-perspective enterprise modeling: Foundational concepts, prospects and future research challenges. online first: http://link.springer.com/article/10.1007/s10270-012-0273-9. *Software and Systems Modeling*.

Frank, U. (2014a). Enterprise modelling: The next steps. *Enterprise Modelling and Information Systems Architectures*, 9(1):24–40.

Frank, U. (2014b). Multilevel modeling: Toward a new paradigm of conceptual modeling and information systems design. *Business and Information Systems Engineering*, 6:accepted for publication.

Frank, U. and Strecker, S. (2009). Beyond erp systems: An outline of self-referential enterprise systems: Requirements, conceptual foundation and design options. icb research report, no. 31, university of duisburg-essen.

Grossmann, R. (1983). *The Categorical Structure of the World*. Indiana University Press, Bloomington.

Kahneman, D., Slovic, P., and Tversky, A., editors (1982). *Judgment under uncertainty: Heuristics and biases*. Cambridge University Press, Cambridge and New York.

Krogstie, J. (2007). Modelling of the people, by the people, for the people. In Krogstie, J., Opdahl, A., and Brinkkemper, S., editors, *Conceptual Modelling in Information Systems Engineering*, pages 305–318. Springer Berlin Heidelberg.

Lakoff, G. (1990). *Women, fire and dangerous things: What categories reveal about the mind*. Univ. of Chicago Press, Chicago, 1 edition.

Lankhorst, M. M. (2005). *Enterprise architecture at work: Modelling, communication, and analysis*. Springer, Berlin and Heidelberg and New York.

Liskov, B. H. and Wing, J. M. (1994). A behavioral notion of subtyping. *ACM Transactions on Programming Languages and Systems*, 16:1811–1841.

Mellor, S. J. (2004). *MDA distilled: Principles of model-driven architecture*. Addison-Wesley object technology series. Addison-Wesley, Boston.

Morin, B., Barais, O., Jézéquel, J.-M., Fleurey, F., and Solberg, A. (2009). Models@run.time to support dynamic adaptation. *IEEE Computer*, 42(10):46–53.

Object Management Group (2006). Meta object facility (mof) core specification: Version 2.0.

Pastor, O. and Molina, J. C. (2007). *Model-Driven Architecture in Practice: A Software Production Environment Based on Conceptual Modeling*. Springer, Berlin and New York.

Proper, E., Lankhorst, M. M., Schönherr, M., Barjis, J., and Overbeek, S., editors (2010). *Trends in Enterprise Architecture Research: 5th Workshop, TEAR 2010,* *Delft, The Netherlands, November 12, 2010, Proceedings*, volume 70 of *Lecture Notes in Business Information Processing*. Springer, Berlin and Heidelberg and New York.

Scheer, A.-W. (1992). *Architecture of Integrated Information Systems: Foundations of Enterprise Modelling*. Springer, Berlin and New York.

Song, H., Huang, G., Chauvel, F., Xiong, Y., Hu, Z., Sun, Y., and Mei, H. (2011). Supporting runtime software architecture: A bidirectional-transformation-based approach. *Journal of Systems and Software*, 84(5):711–723.

Stahl, T. and Völter, M. (2006). *Model-driven software development: Technology, engineering, management*. John Wiley, Chichester and England and Hoboken and NJ.

Wiederhold, G., Wegner, P., and Ceri, S. (1992). Toward megaprogramming. *Commun. ACM*, 35(11):89–99.

A Meta-architecture for Service-oriented Systems and Applications

Leszek A. Maciaszek[1,2], Tomasz Skalniak[1] and Grzegorz Biziel[1]
[1]Wrocław University of Economics, Komandorska 118/120, 53-345 Wrocław, Poland
[2]Macquaire University, Sydney, Australia
leszek.maciaszek@mq.edu.au, {tomasz.skalniak, grzegorz.biziel}@ue.wroc.pl

Keywords: Meta-architecture, Architectural Design, Service-oriented Systems and Applications, System and Software Complexity, Dependency Relationships, Software Quality, Software Adaptability, Holon Abstraction.

Abstract: The paper proposes a new meta-architecture as a reference model for developing service-oriented systems and applications. The seven-layer meta-architecture is called STCBMER (Smart Client - Template - Bean - Controller - Mediator - Entity - Resource). The purpose of it is to reduce software complexity and ensure the quality of adaptability defined as the degree to which an information system or application is difficult to understand, maintain and evolve. The main difficulty stems from complex interactions (dependencies) between system elements. The dependencies can be minimized if the system under development adheres to the architectural design and can be verified by analysing the implementation code. The paper reinforces the proposition that an architectural intent for adaptive complex systems requires some sort of hierarchical layered structure (according to the holon abstraction as an approach to restraining software complexity).

1 INTRODUCTION

The main concern and objective of software architectural design is to manage complexity in resulting systems and applications. Software complexity must not be higher than the complexity of the problem domain addressed by the software. If it is higher, we say that the software solution is over-complex (unnecessarily complicated). The main condition for lowering software complexity is to base its architectural design on a complexity-minimizing architectural framework or reference model (i.e. a meta-architecture).

Complexity is an axiomatic, but relative concept, which can only be properly interpreted by its relation to its contrary notion of simplicity (Agazzi, 2002). Something is complex because it is not simple, and vice versa.

Complexity is also a multi-faceted concept – what is complex from one point of view may be simple from another point of view. In other words, complexity is the combination of several attributes, which need to be examined separately "so that we can understand exactly what it is that is responsible for the overall "complexity". Nevertheless, practitioners and researchers alike find great appeal in generating a single, comprehensive measure to express "complexity"" (Fenton and Pfleeger, 1997).

In our opinion, a complexity measure, if one can be generated, should be seen as an overriding measure of systems and software quality. Therefore, complexity is a derivative of characteristics constituting system/application quality. As noted by Robert Glass (2005) "the task of building quality into software is almost the same as the task of making it maintainable" (or adaptable in our parlance).

The SQuaRE standard (ISO, 2011) identifies eight quality characteristics, of which the quality of maintainability represents the instrumentation side of complexity. The standard identifies further five sub-characteristics of maintainability: modularity, reusability, analysability, modifiability, and testability. We believe that a better term for these sub-characteristics is adaptability (or adaptiveness) rather than maintainability. Adaptability is a broader concept combining understand-ability as a precondition of maintainability and maintainability as a precondition of evolve-ability.

System/software adaptability is underpinned by its complexity, measured as the count of (permitted) dependency relationships in the system/software, where: "A dependency is a relationship that signifies that a single or a set of model elements requires other model elements for their specification or implementation. This means that the complete semantics of the depending elements is either semantically or structurally dependent on the definition of the supplier element(s).”(OMG, 2009).

In our research, we address the last of the five deep questions in computing identified by Jeannette Wing (2008): “(How) can we build complex systems simply?”. We have argued that a valid answer to this question is to construct system/software according to dependency-minimizing meta-architecture (e.g. Maciaszek and Liong, 2005).

The rest of the paper is organized as follows. Section 2 summarizes the PCBMER meta-architecture and makes a case for adjusting and extending it to suit modern service-based systems and applications. Section 3 defines the "service enterprise" viewpoint on complexity and change management in systems and applications. This section introduces a new meta-architecture Smart Client - Template - Bean - Controller - Mediator - Entity - Resource (STCBMER). The meta-architecture refers to the technology-specific frameworks (used and validated on a large e-marketplace project in the domain of Ambient Assisted Living (AAL), but not used here as a case study for the lack of space). The related work, the conclusion and the future work sections close the paper's discussion, and they are followed by the list of references.

2 A RECAP OF THE PCBMER META-ARCHITECTURE FOR ENTERPRISE INFORMATION SYSTEMS

The architecture informs how system/software elements are interlinked. It abstract away from implementation and it omits information not related to interactions between elements. There can be many levels of architectural abstraction. We distinguish between a meta-architecture as a desired holonic structure and concrete instantiations of it in system/software under development. Those concrete instantiations (or architectures) must conform to the chosen meta-architecture so that the complexity-minimization objective is achieved.

A layered, ideally holonic-like structure is the first sine qua non condition for an architectural solution leading to the production of adaptive systems. The PCBMER is our original meta-architectural proposal for such architectural instantiations. The second sine qua non is the use of managerial dependency analysis tools to ascertain adaptability in concrete instantiations. The DSM is our managerial tool of choice for dependency analysis.

An architectural division into layers, apart from complexity reduction, has many other advantages. Without much trouble we can exchange components within a layer, e.g. within the Presentation layer we can change HTML pages to dynamic JSP pages. Moreover, a layer can only communicate with neighbouring layers and only in a single-directional way (i.e. cyclic references are not permitted). As a result, changes in a layer do not require changes in independent layers (i.e. layers that do not depend on the modified layer).

Figure 1 illustrates the PCBMER meta-architecture modelled in UML and showing layers as UML packages. There are six layers: Presentation, Controller, Bean, Mediator, Entity, Resource (e.g. Maciaszek, 2007). Figure 1 shows also Utility Data Sources (typically databases) accessible exclusively from the Resource layer.

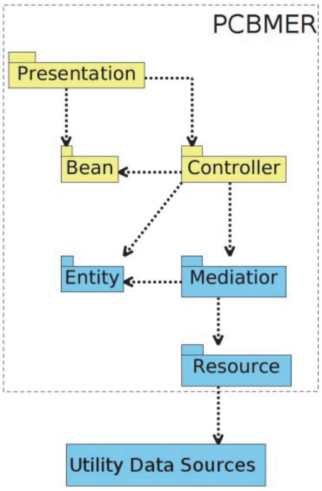


Figure 1: The original PCBMER meta-architecture.

The *Presentation* layer represents the graphical user interface (GUI) objects on which the data (beans) from the Bean layer can be rendered. It is responsible for maintaining consistency in its presentation when the beans change. So, it depends on the Bean layer.

The *Bean* layer represents the data classes and value objects that are destined for rendering on GUI. Unless data is entered by the user, the bean data is built up from the entity objects (the Entity layer).

The *Controller* layer represents the application logic. Controller objects respond to the Presentation requests resulting from user interactions with the system.

The *Entity* layer responds to Controller and Mediator. It contains business objects retrieved from the database or created for successive storage in the database. Many entity classes are container classes (i.e. they contain business objects and methods for adding and removing objects as well as methods to iterate over objects).

The *Mediator* layer mediates between Entity and Resource classes. It manages business transactions, enforces business rules, instantiates business objects in the Entity layer, and in general manages the memory cache of the application. Architecturally, Mediator serves two main purposes. Firstly, to isolate the Entity and Resource layers so that changes in any one of them can be introduced independently. Secondly, to mediate between the Controller and Entity/Resource layers when Controller requests data, but it does not know if the data has previously been loaded from the database into memory.

The *Resource* layer is responsible for all communications with external persistent data sources (databases, web services, etc.). This is where the connections to the database servers are established, queries to persistent data are constructed, and the database transactions are instigated.

The downward arrows between the PCBMER layers signify acyclic dependency relationships. Cyclic dependencies are the main characteristic of over-complex systems and the culprit of the lack of adaptability in such systems. The Downward Dependency Principle (DDP) and the Cycle Elimination Principle (CEP) are two main architectural principles of PCBMER (Maciaszek and Liong, 2005).

The DDP principle ensures that all message dependencies (function calls) have downward direction (message dependencies signify tightly coupled communication, such as in Remote Method Invocation (RMI) - not to be confused with asynchronous messaging, such as in Java Messaging Service (JMS)).

Higher PCBMER layers depend on lower layers, but not vice versa (at least not from the viewpoint of message dependencies). As a result, managing change in lower layers is more troublesome and we need to endeavour to apply extra care to designing lower layers, so that they are more stable (i.e. more resilient to changes).

The DDP principle is further constrained by the Neighbour Communication Principle (NCP). This principle ensures that objects can communicate with distant layers only by utilizing chains of message passing through neighbouring layers. Occasional claims in the literature that such message passing impacts performance are misguided, in particular in the context of enterprise information systems in which performance is invariably related to input/output data transfers to/from databases (performance penalty of in-memory processing is negligible in this context).

The CEP principle demands that cycles of messages are disallowed between objects. The principle applies to objects of any granularity (methods, classes, components, services, packages, subsystems, etc.). This does not mean that call-backs are disallowed. It just means that call-backs must be implemented using other than straight message passing techniques. The two principal techniques are event processing and the use of interfaces, sometimes combined to achieve a desired effect. Additionally, clustering and de-clustering of objects can result in elimination of some cycles. Maciaszek and Liong (2005) contains a detailed description of cycle-elimination techniques.

The Upward Notification Principle (UNP) is a separately-listed principle to counteract the stringent DDP rule and to enforce the CEP principle in communications between layers. This principle requires that lower layers rely on event processing (publish/subscribe protocols) and interfaces to communicate with objects in higher layers.

The PCBMER meta-architectural framework has been created for and validated in development of large scale "stovepipe" enterprise information systems and applications. The software production in such projects is entirely in the hands and minds of the software development team. However, modern software production is not "stovepipe" any more. Software development projects are not standalone undertakings - they are endeavours in systems integration. Complexity management and delivery of adaptable solutions takes on a new dimension.

Firstly, the shift from systems development to systems integration manifests itself on the software level by the shift from synchronous message passing to asynchronous event processing (Maciaszek, 2008a). This has an obvious business explanation. Integration implies dependency on the code that is not our own and not under direct control of the developers (or rather integrators, to be precise). Frequently, this is the code of our business partners who are unlikely to open it up for synchronous message passing from/to our code. But even in case of the integration projects within the same organization, the independent nature of separate business processes (and the software supporting them) is unlikely to permit or warrant synchronous interoperability. Moreover, whether integrating with external systems or with internal systems, synchronous message passing typically would require some level of intervention in the source code of the system we integrate with. Clearly, this is almost never an option.

Secondly, and related to the systems integration issue, another paradigm shift has been observed in modern software production - the shift from in-house software ownership to trusted provisioning of service-based systems and applications. Grounded in the Service Oriented Architecture (SOA) model of computation, this shift has created a new dimension to our understanding of software complexity and delivery of adaptable Software as a Service (SaaS) solutions. The first and foremost concern are the implications for architectural design of such systems and applications. This is discussed next.

3 THE STCBMER META-ARCHITECTURE FOR SERVICE ENTERPRISE

Founded on cloud computing, the SaaS phenomenon exerts new business and pricing models for using information systems without owning them. Service-oriented systems have emerged as a new scientific abstraction allowing orchestration of service resources and processes according to value propositions (co-creation of value).

Service systems and applications have become a commodity - like telephone, water, energy, gas, etc. Associated with this observation, several dichotomies have emerged. On one hand, software products are servitized; on the other hand, software services are productized (Cusumano, 2008). On one hand, vendors of Component of the Shelf (COTS) enterprise information systems use Internet as a service delivery mode; on the other hand, productized services are delivered over Internet as enablers and productivity enhancers in the service economy.

The above dichotomies have posed new challenges on the very idea of complexity and change management in a modern-age service enterprise. The responsibilities for complexity and change management have shifted to producers and suppliers/vendors of service systems and applications, but much of the risk is endured by the enterprises receiving/buying the services. It comes as no surprise that enterprises seek to alleviate the risks and try not to lose control over their own destiny.

The main objective and sine qua non in such service enterprises must be to ensure the adaptability of received service systems and applications. This in turn implies a demand for a layered, modular and dependency-minimizing architecture in such systems and applications, so that the service enterprise can understand, maintain and evolve its software solutions. In this context, it does not matter if a service system or application is delivered as a complete SaaS solution or it is delivered as componentized web services from which a system or application is constructed. In all cases a level of trust between providers and recipients of services is necessary, and in all cases we need to ensure the quality of adaptability in service solutions.

Interestingly, but also paradoxically, the service systems and applications are built on the technologies that, by their very nature, support adaptability. The concepts such as loose coupling, abstraction, orchestration, implementation neutrality, configurability, discoverability, statelessness, immediate access, etc. are exactly the ideas of adaptable architectural design. In the remainder of the paper, we propose a meta-architecture for adaptable architectural design of SOA systems and applications. The meta-architecture has evolved from the PCBMER meta-architecture and it is called Smart Client - Template - Bean - Controller - Mediator - Entity - Resource (STCBMER).

The seven layers of the STCBMER meta-architecture can be grouped into three main architectural modules as shown in Figure 2. The three modules - Smart Client Logic, Application Logic, and Business Logic - work in different address spaces separated by the technology of web services. The SOA technology is responsible for discovering web services, providing service binding, and orchestrating an exchange of information through web service interactions. The service discovery dependencies can be realized through WSDL (Web Services Description Language). The service binding dependencies can be realized through SOAP (Simple Object Access Protocol) or REST (Representational State Transfer) invocations.

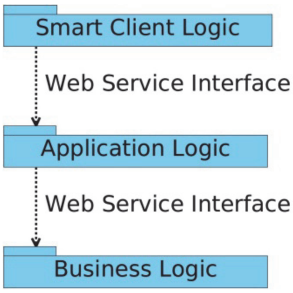


Figure 2: The main modules of the STCBMER meta-architecture.

Figure 3 shows the layered model of the STCBMER meta-architecture. Layers are represented as the UML packages. In the discussion that follows we identify possible technologies for the packages and sub-packages (based on the ones that we have used in a specific instantiation of the meta-architecture in a large project that has served as a validation platform for our architectural vision).

The arrows between the STCBMER packages and sub-packages signify message dependencies. Figure 3 shows also the connectivity from the Smart Client layer to a Web Browser as a typical user interface and the connectivity from Resource to Utility Data Sources.

The most independent and therefore most stable layer is *Resource*. The Resource is a layer responsible for communication with Utility Data Sources (relational databases, NoSQL databases, LDAP directories, etc.). It contains tools to communicate with the database, manage database sessions, construct database queries, etc. Being the most stable layer, it allows easy switching between data sources without making changes in higher layers. The Resource connects to a data source, constructs queries and allows building Entity objects (by Mediator) based on various data sources. The SQL-Alchemy framework is a possible technology for the Resource layer.

The *Entity* layer contains two sub-layers: Entity Object and Entity Object Adapter. The Entity Object package holds business entities, which are mapped (loaded) from data sources. They can be mapped from one or more database tables or views using well known mapping patterns.

ORM (Object-Relational Mapping) frameworks, such as SQL-Alchemy, provide two ways of defining concrete mappers: mapping can be defined as an external class or it can be defined directly in an entity object class. In theory, better and cleaner way is to define the mapper as the external mapping class. In practice, mapping directly in the entity object class may be preferred because in the external mapping all database relationships are added dynamically to the entity object class and are not directly visible in the code as accessible attributes (when for example SQL-Alchemy is used).

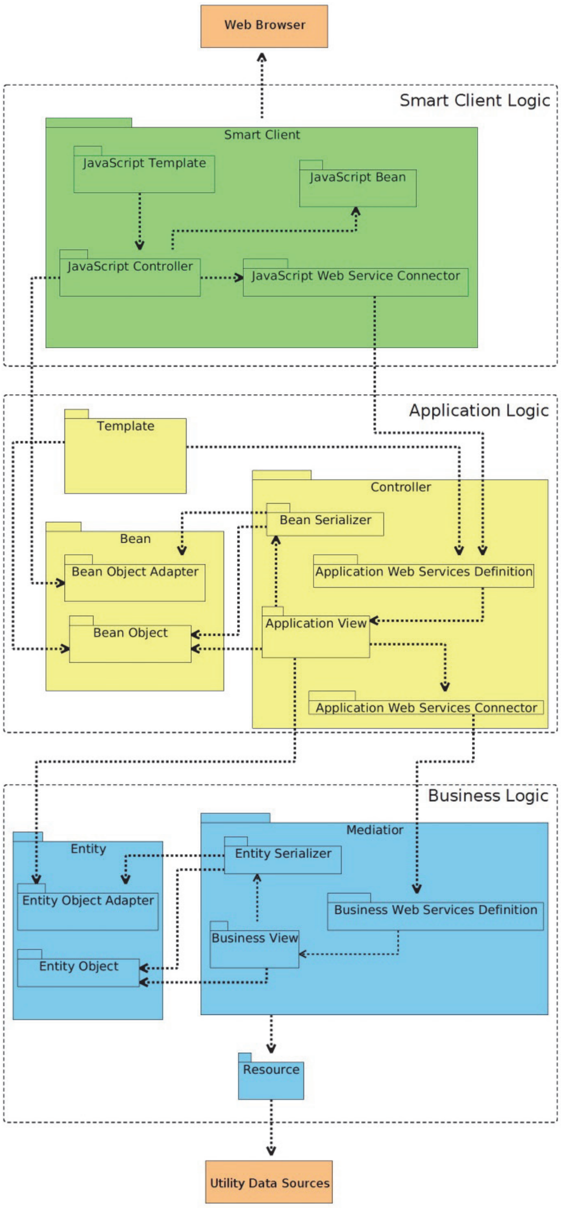


Figure 3: The STCBMER meta-architecture.

The Entity Object Adapter package is a set of classes, which represent entity objects which are serialized and ready to send via a web service. Also every entity object adapter class decides which attributes of the original entity object should be visible to external applications (web services consumers). JSON-based (JavaScript Object Notation) representation might be a good choice, especially if the web service is built with a REST Web Service. JSON is a native JavaScript type, so it suits well web programming, and it is a reasonable alternative to the XML (eXtensible Markup Language).

The *Mediator* layer is responsible for managing business transactions and business rules as well as loading and unloading business objects (entity objects). This layer manipulates entity objects and defines a kind of Facade pattern, which offers access to them: getting, saving, creating, deleting, editing and caching.

As a technology-specific example, the Mediator could use the SQL-Alchemy or other ORM framework to communicate with the Resource layer (which also could be based on the SQL-Alchemy) to load/unload entity objects. Entity objects manipulation (the Mediator) could be available as a set of simple Python functions as well as a set of web services defined with the Pyramid web framework (as in our platform of choice) and accessible via the REST interface. Those functions should be defined in the Business View sub-package and are called “business views”.

To construct a web service (view) from a Python function, a programmer can use a special decorator (Decorator pattern) provided by the Pyramid framework. Since all web methods (views) are available via the REST (Representational State Transfer) interface, every web service should be accessible with a given URI (Uniform Resource Identifier). Routing from a given URI to a specific web service is done by the Pyramid itself. This functionality in the STCBMER meta-architecture is realized by the Business Web Service Definition package (and analogously by the Application Web Service Definition package in the Application Layer).

In the SOAP-based web service the Business Web Service Definition package should also build a WSDL document describing the web API (Application Programming Interface) of the Business Logic. If the API is built as a REST interface, this package should also define the mappers (routes) from a specific URI to a given view (web service). In the Pyramid web framework all the routes are defined by programmers using regex (regular expression) patterns. To serialize and send business objects via the REST interface, the Mediator uses the Entity Serializer. Every serialized entity object is a JSON object, with structure defined in the Entity Object Adapter package.

The *Controller* layer defines the application logic (different from the Mediator's business logic). In our technology-specific scenario, the application logic is captured in a set of functions (Application View sub-package) accessed as pure Python functions or web services. Each function (web service) is called a view (just like in the Mediator layer). The Controller uses the Mediator to get entity object adapters to create and operate on Bean objects. Mapping between the Entity Object Adapter and the Bean Object classes is done by the Application View package. Because the Mediator is accessed via the REST interface, there is a need to cover the REST communication with a Facade component responsible for a networking communication.

The Controller is equipped with the Application Web Service Connector sub-package used by application views to realize the Mediator communication and orchestration. All web services (views) are available through the REST interface. This is why the Controller contains the Application Web Service Definition sub-package, which can be built with the Pyramid framework, and works in the same way as an analogue package in the Mediator layer.

Controller's views return different types of data. Sometimes they pass prepared data to the Template package (bean objects) to get from it an HTML document. Sometimes views provide only pure bean objects in the serialized (JSON) notation (bean object adapters – analogously to entity object adapters). This kind of data can be used by different web services, for example JavaScript Controllers or other applications.

In Figure 3 we present only one application consuming the Business Layer (plus the Smart Client application), but in the STCBMER model the Business Layer can serve the business services (as web services) to more than one application written in various technologies.

The *Bean* layer is just a set of classes that define application objects. Objects of those classes can be used by the Template layer to generate the web front-end (HTML, CSS, eventually JavaScript). But in some cases Bean objects are just returned as a result of invoking an application view (a web service). In this case they are mapped by the Bean Serializer to the bean object adapters. Bean objects are defined dynamically by Controller views and can be stored in JSON notation, which is close to a native type of Dictionary in Python and it is a native type for the JavaScript language. The JSON notation is nowadays widely used in web systems because the text representation of JSON objects (which in the end is sent via HTTP) is quite lightweight and easy to parse in various technologies.

The *Template* layer is responsible for generating a web front-end using Bean objects (prepared by the Controller module in views). While views (Controller) construct data to be displayed, the Template is responsible for how data will be displayed. In our technology-specific solution, the Template layer uses the Mako template library written in Python and is responsible for generating HTML documents (sometimes with some additional CSS and JavaScript, if the documents have to be prepared dynamically). In general the Template layer is used also to generate different types of documents which might be needed by various remote applications/systems.

The *Smart Client* layer consists of the JavaScript Controller, JavaScript Template, JavaScript Bean and JavaScript Web Service Connector. In our e-marketplace project (not described here, as mentioned in passing), all modules except the JavaScript Web Service Connector are provided by the Angular.js framework, which is based on the M-V-VM (Model-View-ViewModel) pattern. This pattern is used by a large number of web frameworks, also by JavaScript frameworks, working usually in a homogeneous memory environment (all objects can access each other).

The ViewModel listens to the Model object (usually as a Subscriber), and after triggering an event, does some application logic (for example changing the state of other Model objects). In the end, the ViewModel can publish its own event object, so that the View (which is usually a Subscriber) could re-render the user interface based on ViewModel attributes (which the ViewModel defines for each Model – similar to the Adapter pattern). Of course, the Angular.js framework is just our platform of choice for the Smart Client layer in our e-marketplace project and it could be realised with different technologies based on various patterns (M-V-VM is just an example).

The STCBMER meta-architecture is an extension of the PCBMER meta-architecture to cater for service-oriented systems and applications. Both meta-architectures share the same complexity-minimizing architectural principles. The four principles discussed earlier (namely CEP, DDP, UNP, and NCP) are all honoured by the STCBMER meta-architecture.

4 RELATED WORK

The word "architecture" is an overloaded term in computing. It is used to denote physical architectural design as well as logical architectural design. In its physical meaning, it refers to the allocation of software components, and communication patterns between them, to computing nodes forming architectural tiers. In its logical meaning (as addressed in this paper), it refers to the allocation of software components, and communication patterns between them, to computing packages forming architectural layers. In between these physical and logical meanings, there are various mixed uses of the word "architecture", including SOA, ADL (Architecture Description Language), Enterprise Architecture, etc.

Although the term "architecture" is overloaded and even overused in the literature, it comes as a surprise that very little research has been reported on layered architectural design for the development of software systems and applications. While complete meta-architectural proposals are difficult to find, the literature is full of architectural guidelines and patterns of which the Core J2EE Patterns (Alur et al., 2003) and the PEAA (Patterns of Enterprise Application Architecture) (Fowler, 2003) have made most impact on our work.

The philosophical underpinning of structuring our models of meta-architectures into hierarchical layers comes from the holonic approach to science as the most promising way to take control over complexity of artificial systems (Koestler, 1967; Koestler, 1980; Capra, 1982; Agazzi, 2002). Apart from dismissing network structures as untenable for construction of complex adaptive systems, the holonic approach explains so called SOHO (Self-regulating Open Hierarchic Order) properties in biological systems. These properties provide a basis for better understanding of human-made systems and how adaptive complex systems should be modelled.

Software complexity underpins all efforts to achieve software quality. Software quality models and standards, such as SQuaRE (ISO, 2011), tend to concentrate on software product quality, but recognize that it is not possible to produce a quality product without having a quality process that defines lifecycle activities. It is in the very nature of software engineering that a major activity within a software quality process is change management.

There is a growing body of research on service change management (e.g. Wang and Wang, 2013), but we do not know of published works that would link change management in service-oriented systems to architectural design as the crux of complexity management and software adaptability.

Similarly with regard to software metrics - a huge number of generic software metrics have been proposed (e.g. Fenton and Pfleeger, 1997). There exist also proposals of metrics targeting service-oriented systems (e.g. Perepletchikov and Ryan, 2011). However, the metrics are not sufficiently linked to the quality assurance processes that would be enforcing architectural design in the software. In other words, the metrics are reactive rather than proactive.

The same observation applies to the DSM method as a visualization of software complexity as well as a vehicle for calculating complexity metrics (Eppinger and Browning, 2012; Sangal et al., 2005). The expressive power of DSM has been mostly used for discovering complexity problems in the software, and for fixing problems like cyclic dependencies, but there is a lack of tangible results reporting round-trip engineering use of DSM to control software complexity.

5 CONCLUSION

The introduction and description of the STCBMER meta-architecture is a contribution of this paper. When we started working on a meta-architecture proposal for service-oriented systems and applications, we expected a notable departure from our PCBMER meta-architecture developed for conventional enterprise systems. It has turned out that STCBMER and PCBMER are similar.

The STCBMER introduces one new layer built with JavaScript and few new sub-packages. A web browser is now an explicit part of the new model.

The Entity and the Bean layers are now defined with more details. Each consists of two sub-packages: one containing the real objects (the Entity Object and the Bean Object) and the second representing objects ready to send via a web service interface (the Entity Object Adapter and the Application Object Adapter).

To map business/application objects to proper adapters, special packages are introduced: the Entity Serializer and the Bean Serializer. Since the communication between the Smart Client Logic, the Application Logic and the Business Logic is organized with a web service technology, special web service packages are introduced. The first type of packages needed to organize a web service communication, are packages which contain the API definition: the Business Web Service Definition and the Application Web Service Definition. Those packages define how the API of each layer looks like. The second type of packages are web service connectors: the JavaScript Web Service Connector and the Application Web Service Connector.

Some differences between STCBMER and the PCBMER can be noticed in dependency relationships. New dependencies exist to reflect the fact that the new meta-architecture works in a web service environment. For example in the PCBMER the direct dependency between Controller and the Entity (Controller's objects construct Bean objects from the Entity objects) is in the STCBMER defined as a dependency between the Controller package and the Entity Object Adapter package. But since the Entity Object Adapter is a sub-package of the Entity package, dependency between the Controller and the Entity layers still exists.

Other differences can also be noticed – not in the architecture definition but in default technical environment. The PCBMER has not been defined to work in a web environment, or in a service-oriented model. The STCBMER is an elaborated version of PCBMER designed to be able to work in those environments.

6 FUTURE WORK

The STCBMER meta-architecture proposed in this paper has been validated in the field on a large project for the e-marketplace domain. However, the usability of the meta-architecture is only a partial proof of its value. In the follow-up research we need to develop concrete metrics that can be used to measure complexity of comparable versions of software designs and systems built according to the STCBMER framework.

The metrics will measure dependency relationships in software. To this aim, we first need to classify all kinds of dependencies in service-oriented systems and applications that have a clear impact on software complexity. At the beginning we will concentrate on coarse-grained dependencies: message dependencies (addressed in this paper, but not in the context of metrics), event dependencies and interface dependencies. For the service-oriented systems and applications, a special attention will need to be placed on the interface dependencies as they constitute the essence of web services. As an important aspect of our future research, we will need to discuss the strengths/weights of various kinds of dependencies on the complexity and adaptability of software.

We stress that the complexity metrics are not absolute measures – their value is only in comparison to other (previous) versions of system/application architectural designs and in successive versions of software products. In Maciaszek (2008b) we discussed the ways of using DSM (Dependency Structure Matrix) for the analysis and comparison of system/software complexity. Today many tools exist that support the DSM method and that additionally integrate with popular IDE-s, such as Eclipse, Visual Studio or IntelliJ.

The tool support is important here as the complexity management has both forward and reverse-engineering dimension. The software needs to be forward-engineered according to its architectural design, but we also need to validate the code conformance with the architectural principles.

Contemporary tools offer visualization of dependencies in the code-base not just at particular levels, such as method-to-method, class-to-class, directory-to-directory, but also across levels, such as function-to-type, namespace-to-class, jar-to-method. One of such tools is Structure101 (Structure, 2014).

Structure101 and most other tools are predominantly reverse engineering tools, more reactive than proactive. Structure101 provides, however, a specialized module, called Architecture Development Environment (ADE), to define architectural rules and guide conformance inside an IDE. The "proactivity" remains at the architecture (instantiation) level and meta-architecture is offered by the tool itself, but we plan to use ADE to define the STCBMER principles for various industrial studies and software development projects.

REFERENCES

Agazzi, E., 2002. What is Complexity? In Agazzi, E., Montecucco, L. (Eds) *Complexity and Emergence. Proceedings of the Annual Meeting of the International Academy of the Philosophy of Science,* pp. 3-11, World Scientific.

Alur, D., Crupi, J., Malks, D., 2003. *Core J2EE Patterns: Best Practices and Design Strategies*, 2nd ed., Prentice Hall.

Capra, F. (1982): *The Turning Point. Science, Society, and the Rising Culture*. Flamingo.

Cusumano, M.A., 2008. *The Changing Software Business: Moving from Products to Services*, IEEE Computer, January, pp.20-27.

Eppinger, S.D., Browning T.R., 2012. *Design Structure Matrix Methods and Applications*, The MIT Press.

Fenton, N.E., Pfleeger, S.L., 1997. *Software Metrics. A Rigorous and Practical Approach,* 2nd ed., PWS Publishing Company.

Fowler, M., 2003. *Patterns of Enterprise Application Architecture*, Addison-Wesley.

Glass, R.L., 2005. The Link Between Software Quality and Software Maintenance. *IT Metrics and Productivity Journal*, November, p.29.

ISO, 2011. *International Standard ISO/IEC 2510: Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - System and Software Quality Models*, ISO/IEC.

Koestler, A., 1980. *Bricks to Babel,* Random House.

Koestler, A., 1967. *The Ghost in the Machine*, Penguin Group, London.

Maciaszek, L.A., 2008a. Adaptive Integration of Enterprise and B2B Applications. In Filipe, J., Shishkov, B., Helfert, M. (Eds), *ICSOFT 2006,* CCIS 10 Springer-Verlag.

Maciaszek, L.A., 2007. An Investigation of Software Holons - The 'adHOCS' Approach. In *Argumenta Oeconomica* Vol.19, No.1-2, pp.1-40.

Maciaszek, L.A., 2008b. Analiza struktur zależności w zarządzaniu intencją architektoniczną systemu (Dependency Structure Analysis for Managing Architectural Intent), In Huzar, Z., Mazur, Z. (Eds), *Inżynieria Oprogramowania – Od Teorii do Praktyki,* pp.13-26, Wydawnictwa Komunikacji i Łączności, Warszawa.

Maciaszek, L.A., 2009. Architecture-Centric Software Quality Management, In Cordeiro, J., Hammoudi, S., Filipe, J. (Eds), *Web Information Systems and Technologies, WEBIST 2008,* LNBIP 18, Springer.

Maciaszek, L.A., 2006. From Hubs Via Holons to an Adaptive Meta-Architecture – the "AD-HOC" Approach. In Sacha, K. (Ed.), *IFIP International Federation for Information Processing, Vol. 227, Software Engineering Techniques: Design for Quality,* pp.1-13, Springer.

Maciaszek, L.A., Liong, B.L., 2005. *Practical Software Engineering. A Case-Study Approach.* Addison-Wesley.

OMG, 2009. *Unified Modeling Language*™ (OMG UML), Superstructure, Version 2.2.

Perepletchikov, M., Ryan, C., 2011: *A Controlled Experiment for Evaluating the Impact of Coupling on the Maintainability of Service-Oriented Software*, IEEE Trans. On Soft. Eng., Vol. 37, No. 4, pp.449-465

Sangal, N. Jordan, E. Sinha, V., Jackson, D., 2005. Using Dependency Models to Manage Complex Software Architecture, In *Procs. OOPSLA'05*, pp.167-176, ACM.

Structure, 2014. *Structure101*, http://structure101.com/, viewed February 2014.

Wang Yi., Wang Ying (2013). A Survey of Change Management in Service-Based Environments, In *SOCA*, pp.259-273, Springer

Wing, J.M., 2008. *Five Deep Questions in Computing*. Comm. of the ACM, Vol. 51, No.1, pp.58-60.

Improving Computer-Support for Collaborative Business Model Design and Exploration

Marin Zec[1], Peter Dürr[2], Alexander W. Schneider[1] and Florian Matthes[1]
[1]*Fakultät für Informatik, Technische Universität München, Boltzmannstr. 3, 85748 Garching, Germany*
[2]*Applied Social Sciences, Munich University of Applied Sciences, Am Stadtpark 20, 81243 München, Germany*
{*marin.zec, alexander.schneider, matthes*}*@tum.de, peter.duerr@hm.edu*

Keywords: Business Model Innovation, Business Model Canvas, Business Model Development, Business Model Exploration, Morphological Analysis, Computer-Aided Business Model Design, Computer-Aided Business Model Generation, Group Dynamics

Abstract: Finding a viable and sustainable business model is a major challenge not only for startup companies. Established companies are re-thinking their existing business models and explore new business opportunities. The Business Model Canvas is currently one of the most popular frameworks for business model innovation. While computer-aided design (CAD) tools are well-established in mechanical engineering, business model design is still mostly done using pen-and-paper methods. In this paper, we (1) discuss benefits and shortcomings of the Business Model Canvas approach, (2) show how it can borrow techniques from General Morphological Analysis to overcome shortcomings, and (3) derive three key requirements for future collaborative CAD tools for business model design. Our analysis contributes to an understanding of how software support can improve collaborative design and evaluation of business models.

1 INTRODUCTION

Business model innovation plays an increasingly important role for both startup as well as mature companies due to increasing competition (Mitchell and Coles, 2003; Mitchell and Coles, 2004). On one hand, the primary organizational goal of startup companies is to generate a viable business model – sometimes resulting in disruption of whole markets. On the other hand, established companies primarily aim to execute their business model as efficiently as possible. As a result, they tend to struggle with rapid and/or profound market changes.

However, many established companies realize the strategic importance of business model innovation for the sustainability of their organization (e.g. Amazon Web Services, a collection of cloud computing services offered by Amazon.com in addition to their primary e-commerce business). Consequently, an increasing number of companies strive for continuous business model innovation. They pursue various strategies to do so, such as promoting and establishing intrapreneurship, corporate venturing or creation of corporate venture capital units. Examples include Google Ventures, Siemens Venture Capital and Unilever Ventures. Still business model innovation remains to be a complex problem for both startups and more mature companies (Chesbrough, 2010).

Business model innovation is typically conducted in teamwork since expertise in different domains such as marketing, engineering and accounting is needed for holistic business model design. However, each expert tends to maintain his/her domain-specific mental model and terminology. Boundary objects make collaboration across different groups possible since they provide a common point of reference for discussion and collaboration. Individuals/groups with different background can interpret them differently without surrendering the shared boundary object's identity (Carlile, 2002). The Business Model Canvas (BMC) (Osterwalder and Pigneur, 2010), an artifact designed to facilitate business model design by providing a problem structure and focus of thought (Eppler et al., 2011), can serve as boundary object for business model innovation.

The BMC was at first proposed as a pen-and-paper or whiteboard tool for business model design workshops. However, Osterwalder et al. (2013) call for CAD software for business modeling since they expect it to yield benefits for strategic planning similar to the benefits and features CAD software brought to engineering or architecture, e.g. "[...] speed, rapid prototyping, quicker visualization, better collaboration, simulation, and better planning [. . .]".

In this paper, we discuss the strengths and shortcomings of state-of-the art BMC software (e.g. facilitation capabilities). We introduce General Morphological Analysis as a powerful technique for complex problem modeling and argue, that the design of CAD software for business model design should borrow useful concepts from computer-aided General Morphological Analysis. The contribution of this paper is the identification of three key requirements for collaborative CAD software for business model design.

2 BUSINESS MODEL CANVAS

The BMC gained popularity in practice since it provides a simple framework and, as a result, facilitates structured discussion about a hypothesized or actual business model. The BMC promotes visual thinking and a holistic perspective on a business model. Visual thinking is a way to develop and clarify ideas about designs and acts as a catalyst for new ideas (Buxton, 2007). Furthermore, part of the power of visuals is their ability to serve as a visible external memory (Baddeley, 1998). Particularly whiteboards are an effective medium for visual thinking due to their free-form nature (Walny et al., 2012).

The BMC is a visual one-page template for describing a business model (Osterwalder and Pigneur, 2010). The BMC comprises four perspectives on a business model: Customers, Offer, Infrastructure, and Financial Viability. Each perspective subsumes one or more business model building blocks; (1) *Customer*: Customer Segments, Customer Relationships, Channels, (2) *Offer*: Value Propositions, (3) *Infrastructure*: Key Activities, Key Resources, Key Partners, and (4) *Financial Viability*: Cost Structure, Revenue Streams. The BMC is comprised of nine building blocks in total (see Figure 1).

The BMC is based on the Business Model Ontology developed in (Osterwalder, 2004). The key idea of the BMC is to provide a simple and common visual framework for communicating and developing business models. The BMC reached wide adoption among practitioners such as business development units, startup companies or seed accelerators.

There is no mandatory sequence according to which the building blocks of the canvas have to be worked out (Fritscher and Pigneur, 2010). Since the building blocks are interrelated, various elements of the business model have to be modified during multiple iterations anyway. However, many practitioners start with identifying customer segments (*Who are we creating value for?*) or the value proposition (*Which value are we creating?*) and iterate over all remaining blocks.

Irrespective of which starting point was chosen, the business model designers scrape through the remaining building blocks in a sequence that seems reasonable to them. They jump back and forth to refine the business model until they are satisfied. During the process, alternative versions can be sketched out on separate canvases or on the same canvas using different colors. The finished BMC represents a business model hypothesis and serves as a basis for subsequent steps in the business model innovation process such as validation or implementation.

In a workshop setting, the BMC can be printed out on a poster or sketched on a whiteboard to provide a shared display. The simple structure and the visual arrangement of the building blocks provide a shared language and point of reference for group discussions.

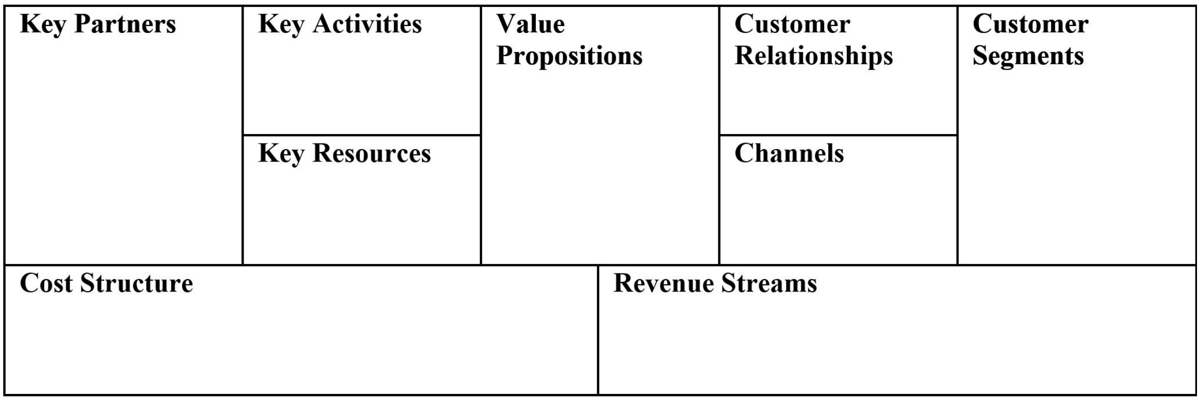
A major advantage of the pen-and-paper or whiteboard approach is that elements (i.e. ideas) can be added, edited, moved around and, if necessary, removed spontaneously. Thus, the current state of the group discussion is captured at any time and group members do not forget ideas in the heat of the discussion. The use of color coding, drawings and rearrangement of elements helps maintain a clear organization of the canvas elements.

2.1 Software Support for BMC

A major drawback of the pen-and-paper as well as whiteboard approach is that the BMC cannot easily be shared with interested stakeholders. In addition, the more elements the canvas contains the harder it is to keep track of them since it gets more and more difficult to maintain clarity.

One way to share a physical BMC is to take a digital picture of it and send the picture to the relevant stakeholder. Another way is to (re)create the canvas using text-processing/graphics/slide-based presentation software or embedding the picture in an electronic document. In any case, creating a BMC using generic software tends to be rather time-consuming (Fritscher and Pigneur, 2010).

Despite a lack of thorough scientific investigation on the effectiveness of BMC mapping software (Eppler et al., 2011), various BMC modeling tools emerged for computer-aided business model design to facilitate easier sharing while providing specific modeling facilities well adapted for the BMC. CAD software for business model design based on the BMC ranges from rather lightweight browser-based canvas tools such as BM|DESIGN|ER



Key Partners

Key Activities

Key Resources

Value Propositions

Customer Relationships

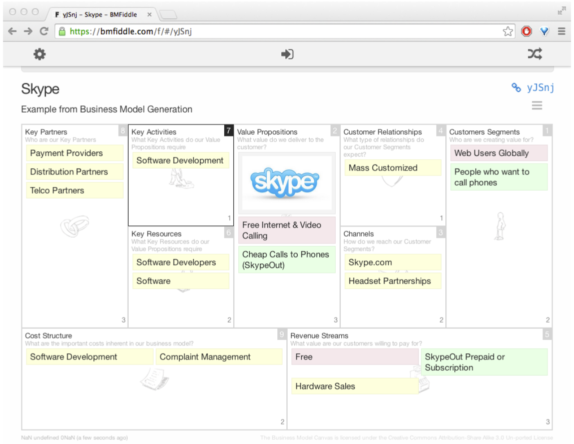
Channels

Customer Segments

Cost Structure

Revenue Streams

Figure 1: The Business Model Canvas (Osterwalder and Pigneur, 2010).



Skype

Example from Business Model Generation

Key Partners 8
Who are our Key Partners
Payment Providers
Distribution Partners
Telco Partners
3

Key Activities 7
What Key Activities do our Value Propositions require
Software Development
1

Key Resources 6
What Key Resources do our Value Propositions require
Software Developers
Software
2

Value Propositions 2
What value do we deliver to the customer?
Free Internet & Video Calling
Cheap Calls to Phones (SkypeOut)
3

Customer Relationships 4
What type of relationships do our Customer Segments expect?
Mass Customized
1

Channels 3
How do we reach our Customer Segments?
Skype.com
Headset Partnerships
2

Customers Segments 1
Who are we creating value for?
Web Users Globally
People who want to call phones
2

Cost Structure 9
What are the important costs inherent in our business model?
Software Development
Complaint Management
2

Revenue Streams 5
What value are our customers willing to pay for?
Free
SkypeOut Prepaid or Subscription
Hardware Sales
3

NaN undefined 0NaN (a few seconds ago)

The Business Model Canvas is licensed under the Creative Commons Attribution-Share Alike 3.0 Un-ported License

Figure 2: Screenshot of Business Model Fiddle (Steenkamp, 2012).

(Fritscher and Pigneur, 2010), Business Model Fiddle – Fig. 2(Steenkamp, 2012), Canvanizer (Proud Sourcing GmbH, 2011) and Strategyzer (Business Model Foundry GmbH, 2013) over meta-model-based wizards (Hauksson and Johannesson, 2014) to rich desktop software suites, e.g. BiZZdesign Architect (BiZZdesign, 2004).

Eppler et al. (2011) compared the use of generic slide-based presentation software to BMC mapping software. Their study found a positive effect of the BMC on perceived collaboration and a negative effect on perceived creativity. The authors conclude “that artefacts can have considerable power in shaping group interactions and idea generation in the context of business model innovation” and call for further research on visual artefacts used to facilitate business model innovation.

2.2 Benefits of BMC Software

Particularly collaborative, web-based BMC software offers two major advantages over pen-and-paper/whiteboard sessions and general-purpose software:

1. **Support for Distributed Teams.** Web applications facilitate collaboration across time, location and organizational boundaries. Since teams are often distributed in terms of at least one of these dimensions, software-based business modeling saves costs in comparison to pen-and-paper/whiteboard workshops.
2. **Easier and More Flexible Customization, Reuse and Sharing.** Canvas elements can be easily customized in terms of coloring, typography and position. Various media can be embedded (e.g. images, video or spreadsheets). Canvases can be saved in various formats, re-opened for further refinement and easily shared with stakeholders.

We analyzed three popular web-based BMC modeling tools: Business Model Fiddle, Canvanizer, and Strategyzer. All tool sticks to the one-page layout of the paper template for their main view. The tools differ slightly in the way how building blocks can be filled with content and how elements can be customized. Some tools offer features which extend the original idea of the BMC. For instance, Strategyzer provides a financial estimator. Strategyzer and Canvanizer offer real-time collaboration, i.e. concurrent editing and chat support. Business Model Fiddle allows many customization of the canvas (e.g. renaming of building blocks) and the creation of snapshots to record changes. In addition, designers can sketch on uploaded images and assign them to building blocks.

The main benefits of CAD for the BMC approach is the support for collaboration across time, location and organizational boundaries as well as reuse. The analyzed tools mimic a paper/whiteboard and exploit benefits of information technology. However, there is little facilitation support since in both, on-site meetings and asynchronous distributed modeling sessions, many methodological questions such as finding a reasonable starting point have to be answered by the modelers. None of the analyzed tools guides business modelers through the process. While sticking to the one-page layout, they slightly differ in the way how building blocks can be filled with content and elements be customized.

2.3 BMC and Facilitation

The role of the BMC in business model design is similar to the notion of grammatical design which constitutes boundaries within which designers can find creative solutions if they deviate from standard solutions (Brown and Cagan, 1996). The BMC frames the discussion and predetermines the general structure of the boundary object. BMC designers tend to "think 'within' the given domains of the template" (Eppler et al., 2011). They are supposed to be creative within said domains. However, there is little guidance on how to make use of one's full creative potential. In practice, groups often ideate through spontaneous associations between concepts. Thus, the business model design process that supplements the BMC is intentionally informal and generic. It consists of five phases which do not necessarily have to be gone through in a linear manner (Osterwalder and Pigneur, 2010): (1) *Mobilize* (i.e. preparation), (2) *Understand* (i.e. research and analyze the context), (3) *Design* (i.e. generation of business model hypotheses using the canvas), (4) *Implement* (i.e. implementing the best hypothesis in the field), and (5) *Manage* (i.e. monitor the market and update the business model accordingly). The process model does not prescribe what to do in a specific phase. Instead, the authors refer to various other tools and methods.

Business model generation demands a broad range of skills, knowledge and experience as well as creativity. Thus, business model generation is often conducted in teamwork. There is a widespread belief that the performance of interactive groups is higher than the performance of nominal groups (i.e. the aggregate performance of the same number of non-interacting individuals). However, this does not necessarily hold true. For instance, various studies on group performance in brainstorming have shown that nominal groups outperform interactive groups, e.g. (Diehl and Stroebe, 1987; Mullen et al., 1991). While teams tend to generate more ideas if they follow Osborn's brainstorming rules (Osborn, 1957) than when they do not stick to those rules, they are still not as effective as nominal groups. Explanations for this phenomenon include social loafing (Paulus et al., 1993), social anxiety (Camacho and Paulus, 1995) and production block (Diehl and Stroebe, 1987).

While the BMC deserves merit for providing a common language for group discussion, teams have to be careful. Pitfalls in group brainstorming are only one example illustrating the intricacy of group dynamics in collaborative settings.

Teamwork can yield group process gains as well. For instance, team members might become more motivated when facing social competition. They might also improve their skills due to knowledge transfer. And even coordination gains are possible (e.g. conductor of an orchestra).

Generally, process gains as well as losses can occur in terms of motivation, capabilities and coordination. A skilled and experienced facilitator promotes process gains and mitigates process losses. For instance, they play devil's advocate to encourage alternative or counter-intuitive thoughts and, consequently, avoid groupthink and shared information bias (Baker, 2010). However, teams often do not involve a skilled facilitator because it is considered too expensive, they think there is no need for a facilitator (since negative effects are implicit and thus hard to notice) or there is simply no facilitator available.

In many situations, collaborative web-based CAD software is expected to be used without a dedicated facilitator, mostly to save time and money. Therefore, we argue that collaborative BMC software has to compensate for the lack of a skilled facilitator as much as possible. We expect CAD software for business design to enforce meta-model constraints and validate the model (e.g. impose mappings between value propositions and customer segments). This way, the BMC modeling tool can provide, for instance, helpful directions in the case of meta-model violations. Hauksson et al. (2014) implemented a desktop-based wizard for business model design which enforces BMC meta-model constraints.

While it is unlikely that software can fully replace a human facilitator, there are some simple yet powerful techniques to improve collaboration and ideation in groups. For instance, gamification techniques can help mitigate motivation losses. Allowing anonymous contributions might reduce social anxiety or evaluation apprehension and, as a result, increase diversity of ideas since ideas can be expressed uninhibitedly. Separation between divergent thinking (deferring judgment, creating as many ideas as possible) and convergent thinking (think and evaluate ideas analytically), shown to improve creativity processes and results (Cropley, 2006), can be supported by implementing different views and means of interaction for each of those phases. In addition, differentiating between single and team phases might increase the quantity of ideas: during single phases contributions from other team members should be hidden to avoid production block and mutual influence, during team phases contributions from others should be visible such that they can serve as an inspiration for additional ideas and refinements. While we do not provide an exhaustive list of design recommendations, we argue for the integration of insights from social psychology and creativity research into the design of future CAD tools for business modeling.

2.4 Discussion

In general, the BMC captures one specific business model. While it is possible to sketch more business models on the same canvas at the same time, the canvas tends to get confusing. Therefore, in practice, each business model is usually sketched out on a separate canvas.

The BMC facilitates a structured discussion about a business model but it does not guide its users to explore the formal solution space systematically. Rather, users tend to start with specifying one building block and then scrape from building block to building block. However, this heuristic approach might miss innovative and viable solutions. One method to circumvent this disadvantage and to explore the full solution space is Morphological Analysis which is presented in the next section.

Another issue that arises in collaborative settings which is not addressed by the BMC is group dynamics. Social psychology literature has shown how various negative effects such as social loafing tend to occur in group settings. As a result, team performance and productivity might degrade considerably.

3 USEFUL CONCEPTS FROM MORPHOLOGICAL ANALYSIS FOR BMC SOFTWARE DESIGN

In this section, we introduce General Morphological Analysis (GMA), a generic problem modeling technique, and associated software tools. We argue that they feature useful concepts which should be considered to be adopted by CAD software for business modeling. In a broader sense, Morphological Analysis (MA) is concerned with the study of form, structure and interconnections between structural elements (Shurig, 1986). Various disciplines such as linguistics or biology conduct subject-specific variants of MA. Swiss Astronomer Fritz Zwicky developed a generic type of MA commonly referred to as General Morphological Analysis (GMA) (Ritchey, 1998). GMA is a method which aims to facilitate system or problem understanding and structuring. GMA is particularly suited for multi-dimensional, non-quantifiable problems for which mathematical or causal modeling is not applicable or appropriate (Ritchey, 1998). The general idea of GMA is to derive a non-quantified model of the system or problem under examination by identifying its key structural components. In the following, we will focus on GMA for problem structuring and solving.

Parameter A	**Parameter B**	**Parameter C**	**Parameter D**
a1	b1	c1	d1
a2	b2	c2	d2
a3	b3		d3

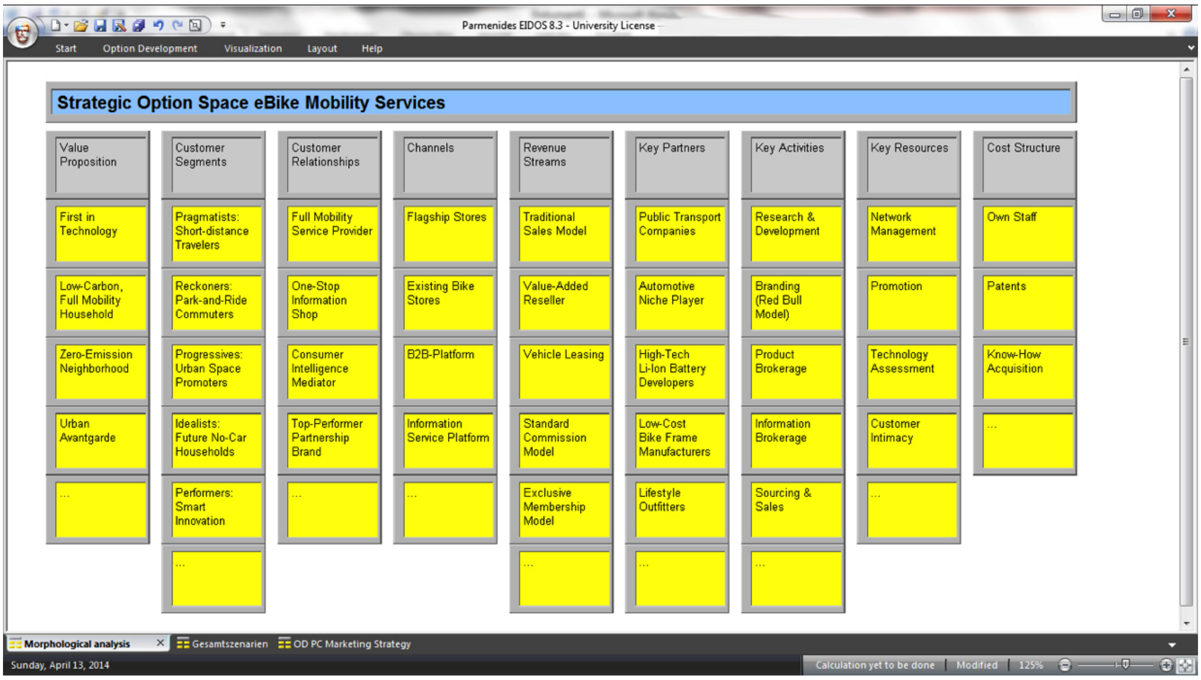
Figure 3: The general structure of a morphological field. One specific formal configuration is highlighted in gray.

3.1 Method

The initial step of GMA is to clarify the problem statement such that there is a shared understanding of the problem. The result of this step is a set of aspects, or parameters, that seem to be the most relevant characteristics of the problem at hand.

GMA distinguishes between parameters (also: components or dimensions) and parameter values. The first step of GMA is to break down the problem into subcomponents. Ideally, the subcomponents are mutually exclusive and collectively exhaustive. In practice, there might be some overlap between parameters. However, the overlap should be as small as possible. Team members have to discuss and collaboratively devise a set of parameters which captures all key aspects of the problem at hand. This step fosters a shared understanding of the problem because the group needs to reach a consensus about what the key aspects of the issue are. For business model design, the BMC provides an established problem structure (i.e. the nine building blocks).

Once an adequate set of parameters is found, the range of parameter values has to be specified for each parameter. Parameter values can be qualitative or quantitative. The level of abstraction is specific to the concrete problem and purpose of the analysis. What is important in this step of GMA is to investigate each parameter independently. This approach promotes di-



Strategic Option Space eBike Mobility Services

Value Proposition	Customer Segments	Customer Relationships	Channels	Revenue Streams	Key Partners	Key Activities	Key Resources	Cost Structure
First in Technology	Pragmatists: Short-distance Travelers	Full Mobility Service Provider	Flagship Stores	Traditional Sales Model	Public Transport Companies	Research & Development	Network Management	Own Staff
Low-Carbon, Full Mobility Household	Reckoners: Park-and-Ride Commuters	One-Stop Information Shop	Existing Bike Stores	Value-Added Reseller	Automotive Niche Player	Branding (Red Bull Model)	Promotion	Patents
Zero-Emission Neighborhood	Progressives: Urban Space Promoters	Consumer Intelligence Mediator	B2B-Platform	Vehicle Leasing	High-Tech Li-Ion Battery Developers	Product Brokerage	Technology Assessment	Know-How Acquisition
Urban Avantgarde	Idealists: Future No-Car Households	Top-Performer Partnership Brand	Information Service Platform	Standard Commission Model	Low-Cost Bike Frame Manufacturers	Information Brokerage	Customer Intimacy	...
...	Performers: Smart Innovation	...	...	Exclusive Membership Model	Lifestyle Outfitters	Sourcing & Sales	...	
	...			...	...	...		

Figure 4: A screenshot from Parmenides EIDOS showing a morphological field which represents the strategic option space for an eBike Mobility Service.

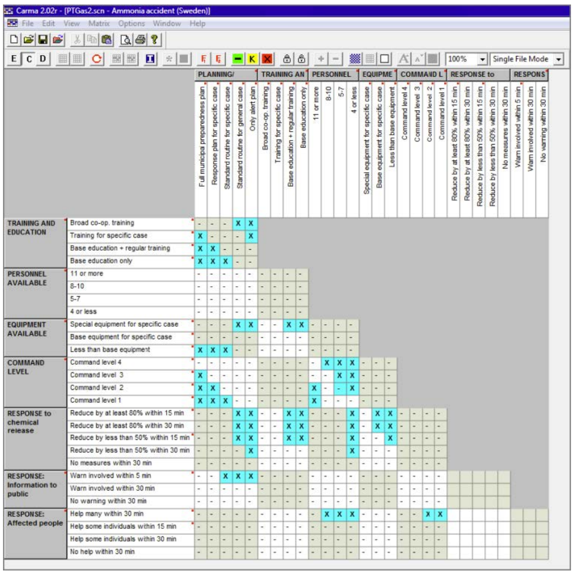
vergent thinking and openness for counter-intuitive values. The focus at this point is on formal properties. Evaluation of feasibility should be deferred. We argue that this is a major advantage over the BMC method because individuals are tempted to think about different building blocks at the same time because they instinctively try to interrelate different building blocks (e.g. ”‘if we want to sell our product to young people we have to build a mobile platform as a channel”’). While this approach can make sense, innovative ideas might not be taken into consideration because the discussion is centering around familiar business model patterns. Thus, business modelers constrain their creativity and, consequently, might miss particularly innovative business model designs.

Parameters and their value ranges constitute a morphological field (MF) which is usually represented by a matrix (see Figure 3). The MF matrix is a dense representation of the formal configuration space. The formal configuration space is the set of all morphological configurations (i.e. parameter value combinations containing one and only one value per parameter). Depending on the perspective and purpose of the analysis, the formal configuration space is sometimes referred to as formal problem space or formal solution space. To sum up, GMA is a structured problem modeling method which enables its users to establish the space of all formal solutions, systematically discuss the contained formal configurations and identify the best solution.

GMA prescribes clear separation between divergent thinking and convergent thinking. By contrast, in BMC workshops, participants tend to mix both styles of thinking too frequently and, as a result, do not systematically explore the space of bounded creativity.

3.2 Example

GMA can be used for various purposes such as scenario analysis, product innovation or strategy development. Figure 4 shows an example for a MF representing the strategic option space for an eBike Mobility Service in terms of BMC terminology. Each parameter (i.e. “building block”, gray background) can take a value from its parameter range (depicted in yellow or white, respectively). A formal business model is given by a specific configuration from the morphological field (i.e. solution space). For instance, a specific formal business model is given by (“First in Technology”, “Pragmatists: Short-distance Travelers”, “One-Stop Information Shop”, “Flagship Stores”, “Vehicle Leasing”, “Public Transport Companies”, “Research & Development”, “Promotion”, “Patents”). Not all possible configurations represent viable business models since some business model elements might be incompatible. However, using GMA the complete set of formal solutions can be identified and systematically evaluated. GMA software can help


Carma 2.02r - [PTGas2.scn - Ammonia accident (Sweden)]
File Edit View Matrix Options Window Help
E C D
100%
Single File Mode

		PLANNING/					TRAINING AN				PERSONNEL				EQUIPME			COMMAND L				RESPONSE to					RESPONS		
		Full municipa preparedness plan	Response plan for specific case	Standard routine for specific case	Standard routine for general case	Only alert plan	Broad co-op. training	Training for specific case	Base education + regular training	Base education only	11 or more	8-10	5-7	4 or less	Special equipment for specific case	Base equipment for specific case	Less than base equipment	Command level 4	Command level 3	Command level 2	Command level 1	Reduce by at least 80% within 15 min	Reduce by at least 80% within 30 min	Reduce by less than 50% within 15 min	Reduce by less than 50% within 30 min	No measures within 30 min	Warn involved within 5 min	Warn involved within 30 min	No warning within 30 min
TRAINING AND EDUCATION	Broad co-op. training	-	-	-	X	X																							
	Training for specific case	X	-	-	-	X																							
	Base education + regular training	X	X	-	-	-																							
	Base education only	X	X	X	-	-																							
PERSONNEL AVAILABLE	11 or more	-	-	-	-	-	-	-	-	-																			
	8-10	-	-	-	-	-	-	-	-	-																			
	5-7	-	-	-	-	-	-	-	-	-																			
	4 or less	-	-	-	-	-	-	-	-	-																			
EQUIPMENT AVAILABLE	Special equipment for specific case	-	-	-	X	X	-	-	X	X	-	-	-	-															
	Base equipment for specific case	-	-	-	-	-	-	-	-	-	-	-	-	-															
	Less than base equipment	X	X	X	-	-	-	-	-	-	-	-	-	-															
COMMAND LEVEL	Command level 4	-	-	-	-	-	-	-	-	-	-	X	X	X	-	-	-												
	Command level 3	X	-	-	-	-	-	-	-	-	-	-	X	X	-	-	-												
	Command level 2	X	X	-	-	-	-	-	-	-	X	-	-	X	-	-	-												
	Command level 1	X	X	X	-	-	-	-	-	-	X	-	-	-	-	-	-												
RESPONSE to chemical release	Reduce by at least 80% within 15 min	-	-	-	X	X	-	-	X	X	-	-	-	X	-	X	X	-	-	-	-								
	Reduce by at least 80% within 30 min	-	-	-	X	X	-	-	X	X	-	-	-	X	-	X	X	-	-	-	-								
	Reduce by less than 50% within 15 min	-	-	-	X	X	-	-	X	X	-	-	-	X	-	-	X	-	-	-	-								
	Reduce by less than 50% within 30 min	-	-	-	-	X	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-								
	No measures within 30 min	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-								
RESPONSE: Information to public	Warn involved within 5 min	-	-	X	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-								
	Warn involved within 30 min	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-								
	No warning within 30 min	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-								
RESPONSE: Affected people	Help many within 30 min	-	-	-	-	-	-	-	-	-	-	X	X	X	-	-	-	-	-	X	X								
	Help some individuals within 15 min	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-								
	Help some individuals within 30 min	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-								
	No help within 30 min	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-								

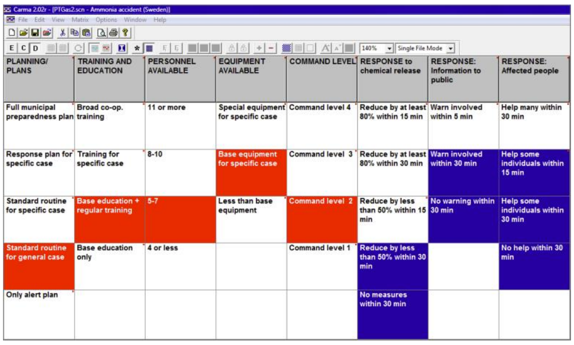
Figure 5: Consistency matrix in MA/Carma. Adopted from (Ritchey, 2005).

reduce the formal solution space to a viable solution space using various techniques such as cross consistency assessment or clustering.

3.3 Sofware Support for GMA

Conducting a GMA by hand has one major challenge: the size of the solution space grows exponentially with each additional parameter. As a result, it is often not possible to evaluate all formal solutions of the MF (e.g. a 5-parameter MF with 6 possible values for each parameter yields $6^5 = 7776$ formal solutions. The example in Figure 4 contains 480.000 formal business models.). Therefore, in workshops without software support, the solution space cannot be analyzed exhaustively. Rather, only a small subset of configurations is selected for deeper analysis according to subjective preferences and within objective constraints (e.g. time limit). GMA software addresses the challenge of large solution spaces by providing means for reduction of the formal configuration space to a practical solution space. Examples for software which supports GMA and solution space reduction include Parmenides EIDOS (Parmenides Foundation, 2014) and MA/Carma (Ritchey, 2005).

MA/Carma allows the creation of an inference model. First, the user has to specify the consistency (or compatibility) of each pair of parameter values. Given such a consistency matrix (see Figure 5), MA/Carma generates an interactive inference model (see Figure 6). The user can declare arbitrary parameter values to be exogenous (colored red or medium gray, respectively) and the software calculates which values are still viable for the remaining parameters (colored blue or dark gray, respectively). A detailed description can be found in (Ritchey, 2005). Parmenides EIDOS supports a similar technique: users have to specify a numeric consistency value for each pair of parameters. Then, a consistency value for each configuration is calculated by the software. As a result, in their subsequent analysis, users can focus on configurations which yield the highest consistency values. In contrast to MA/Carma, Parmenides EIDOS supports clustering of similar configurations to identify more abstract patterns among viable solutions.


Carma 2.02r - [PTGas2.scn - Ammonia accident (Sweden)]
File Edit View Matrix Options Window Help
E C D
140%
Single File Mode

PLANNING/ PLANS	TRAINING AND EDUCATION	PERSONNEL AVAILABLE	EQUIPMENT AVAILABLE	COMMAND LEVEL	RESPONSE to chemical release	RESPONSE: Information to public	RESPONSE: Affected people
Full municipal preparedness plan	Broad co-op. training	11 or more	Special equipment for specific case	Command level 4	Reduce by at least 80% within 15 min	Warn involved within 5 min	Help many within 30 min
Response plan for specific case	Training for specific case	8-10	Base equipment for specific case	Command level 3	Reduce by at least 80% within 30 min	Warn involved within 30 min	Help some individuals within 15 min
Standard routine for specific case	Base education + regular training	5-7	Less than base equipment	Command level 2	Reduce by less than 50% within 15 min	No warning within 30 min	Help some individuals within 30 min
Standard routine for general case	Base education only	4 or less		Command level 1	Reduce by less than 50% within 30 min		No help within 30 min
Only alert plan					No measures within 30 min		

Figure 6: An interactive inference model constructed by MA/Carma. Adopted from (Ritchey, 2005).

3.4 Discussion

GMA is a simple problem structuring method for individuals and groups. It can be used for various problems such as product innovation, strategy development or scenario analysis. MA is particularly useful if the problem at hand cannot be adequately expressed in a mathematical model, many different stakeholders are involved and various viewpoints have to be considered.

MA aims at constructing a formal solution space. Formal solutions in the solution space might be inconsistent and/or impractical. Solution space reduction techniques such as cross consistency assessment in combination with complementary software support help rule out inconsistent configurations and reduce the solution space significantly. This way, the solution space can be explored exhaustively yet efficiently.

To the best of our knowledge, there is no GMA software that is inherently collaborative and supports distributed teams. While GMA software such as Parmenides EIDOS or MA/Carma is used by facilitators to provide a shared display in workshops, their user interface and interaction design is tailored to individual users.

Moreover, we have not found any GMA software that accounts for psychological aspects and group dynamics. However, we argue that BMC software design can borrow from GMA and respective software. They key advantage of the software-based GMA approach to modeling of complex problems such as business model innovation is the systematic construction of the formal solution space and leveraging software to find viable solutions within the practical solution space.

A key feature of MA/Carma is the construction of an inference model of the solution space. Designers can interact with the model and analyze the dependencies of strategic decisions by treating specific parameter values as fixed input. Given a consistency matrix, the software automatically calculates which options remain to be viable for all other parameters. We argue that such an inference model provides more insights into the business modeling space than the depiction of the BMC. The BMC is a descriptive model that only represents one particular business model (or multiple descriptive models if colors are used.).

4 KEY REQUIREMENTS FOR FUTURE BMC SOFTWARE

In the previous sections, we identified three key requirements for future CAD software for business model design: (1) the ability to perform an interactive "what-if" analysis (*inference capability*), (2) methodical guidance throughout the design process (*facilitation capability*) to mitigate negative effects of group dynamics, and (3) the ability to collaborate across time, location and organizational boundaries (*support of distributed teams*).

The inference capability enables designers to experiment with business models by declaring particular business model elements as exogenous and analyzing the implications for the remaining building blocks. The facilitation capability offers guidance during the design process in order to increase creativity of the designers while limiting negative effects stemming from group dynamics. In addition, borrowing techniques from GMA software, future BMC software might provide a concrete process model for systematic generation of business models within the conceptual boundaries of the BMC while still allowing and promoting creativity (i.e. parameter (value) definition, cross consistency assessment and configuration space reduction). The support of distributed teams refers to the ability to carry out the business model design process cost-efficiently in a distributed setting.

Collaborative, web-based CAD software supporting the BMC is readily available. Thus, there is BMC support for distributed teams. However, those tools lack inference and facilitation capabilities.

Therefore, we propose to extend BMC software by borrowing concepts from GMA and respective software. Neither BMC methodology nor GMA address potential group process losses such as motivation loss (e.g. social loafing) or skill impairment (e.g. social anxiety). Thus, groups might not exploit their full cognitive and creative potential.

We argue that methods as well as tools for collaborative business model generation have to address negative effects of teamwork and try to mitigate them. Skilled human facilitators might accomplish this task. However, assessing the skill of a human facilitator is hard. In addition, in some contexts there is no possibility to hire an experienced facilitator or it might be considered too expensive.

The absence of a facilitator can be compensated in BMC software by implementing various facilitation and creativity techniques. For instance, creativity can be increased by clearly separating between divergent and convergent thinking phases. Alternating between individual and team ideation helps decreasing negative impact of anchoring and groupthink. Anonymity can help reduce social anxiety. On the other hand, anonymity might increase social loafing since contributions cannot be attributed to individuals and, thus, individuals might "'hide"' behind the team. Facilitation capabilities have to be well-conceived and evaluated.

5 CONCLUSION AND FUTURE WORK

We identified benefits and shortcomings of state-of-the art CAD software for the BMC methodology. We agree with Osterwalder, Fritscher et al. that CAD software for business model design is likely to improve strategic planning, particularly for distributed business model design teams which face increasing competition (Osterwalder and Pigneur, 2013; Fritscher and Pigneur, 2014). The BMC deserves merit for providing a common (visual) language for business model design. However, we think that CAD support for business model design is still only in its infancy and that various questions need to be answered about how to design effective next-generation CAD software for collaborative business model design. We expect future BMC software to leverage the potential of software support and provide additional value that goes beyond mere digitization of the BMC approach.

Our next step is to refine the general requirements identified in this paper. Then, we will build a prototype which implements the ideas discussed above (e.g. inference and facilitation capabilities) and validate our hypotheses.

REFERENCES

Baddeley, A. (1998). Recent developments in working memory. *Current opinion in neurobiology*, 8(2):234–238.

Baker, D. F. (2010). Enhancing group decision making: An exercise to reduce shared information bias. *Journal of Management Education*, 34(2):249–279.

BiZZdesign (2004). BiZZdesign Architect. http://www.bizzdesign.com. [Online; accessed April 15, 2014].

Brown, K. N. and Cagan, J. (1996). Grammatical design and bounded creativity. Carnegie Mellon University, Department of Mechanical Engineering.

Business Model Foundry GmbH (2013). Strategyzer. http://www.strategyzer.com. [Online; accessed April 15, 2014].

Buxton, B. (2007). *The Anatomy of Sketching*. Morgan Kaufmann, San Francisco and CA and USA.

Camacho, L. M. and Paulus, P. B. (1995). The role of social anxiousness in group brainstorming. *Journal of Personality and Social Psychology*, (68):1071–1080.

Carlile, P. R. (2002). A pragmatic view of knowledge and boundaries: Boundary objects in new product development. *Organization Science*, 13(4):442–455.

Chesbrough, H. (2010). Business model innovation: Opportunities and barriers. *Long Range Planning*, 43(2-3):354–363.

Cropley, A. (2006). In praise of convergent thinking. *Creativity Research Journal*, 18(3):391–404.

Diehl, M. and Stroebe, W. (1987). Productivity loss in brainstorming groups: Toward the solution of a riddle. *Journal of Personality and Social Psychology*, 53(3):497–509.

Eppler, M. J., Hoffmann, F., and Bresciani, S. (2011). New business models through collaborative idea generation. *International Journal of Innovation Management*, 15(6):1323–1341.

Fritscher, B. and Pigneur, Y. (2010). Supporting business model modelling: A compromise between creativity and constraints. In *Task Models and Diagrams for User Interface Design, 8th International Workshop, Revised Selected Papers*, pages 28–43, Brussels and Belgium.

Fritscher, B. and Pigneur, Y. (2014). Computer aided business model design: Analysis of key features adopted by users. In *47th Hawaii International Conference on System Science*, pages 3929–3938, Waikoloa and HI and USA.

Hauksson, H. and Johannesson, P. (2014). Metamodeling for business model design. facilitating development and communication of business model canvas (bmc) models with an omg standards-based metamodel. In *The 8th International Workshop on Value Modeling and Business Ontology (VMBO 2014)*, Berlin and Germany.

Mitchell, D. W. and Coles, C. B. (2003). The ultimate competitive advantage of continuing business model innovation. *Journal of Business Strategy*, 24(5):15–21.

Mitchell, D. W. and Coles, C. B. (2004). Establishing a continuing business model innovation process. *Journal of Business Strategy*, 25(3):39–49.

Mullen, B., Johnson, C., and Salas, E. (1991). Productivity loss in brainstorming groups: A meta-analytic integration. *Basic and Applied Social Psychology*, (12):3–23.

Osborn, A. F. (1957). *Applied imagination: Principles and Procedures of Creative Problem Solving*. Charles Scribner's Sons, New York and NY and USA.

Osterwalder, A. (2004). *The business model ontology: A proposition in a design science approach*, volume HEC 173. Institut d'Informatique et Organisation, Lausanne and Switzerland.

Osterwalder, A. and Pigneur, Y. (2010). *Business Model Generation: a handbook for visionaries, game changers, and challengers*. Jon Wiley & Sons.

Osterwalder, A. and Pigneur, Y. (2013). Designing business models and similar strategic objects: The contribution of is. *Journal of the Association for Information Systems*, 14(5):237–244.

Parmenides Foundation (2014). Parmenides EIDOS. https://www.parmenides-foundation.org/application/. [Online; accessed April 15, 2014].

Paulus, P. B., Dzindolet, M. T., Poletes, G., and Camacho, L. M. (1993). Perception of performance in group brainstorming: The illusion of group productivity. *Personality and Social Psychology Bulletin*, (19):78–89.

Proud Sourcing GmbH (2011). Canvanizer. http://www.canvanizer.com. [Online; accessed April 15, 2014].

Ritchey, T. (1998). Fritz Zwicky, Morphology and Policy Analysis. In *16th Euro Conference on Operational Analysis*, Brussels.

Ritchey, T. (2005). MA/Carma: Advanced Computer Support for General Morphological Analysis. http://www.swemorph.com/macarma.html. [Online; accessed April 15, 2014].

Shurig, R. (1986). Morphology: A knowledge tool. *Systems Research*, 3:9–19.

Steenkamp, J. (2012). Business Model Fiddle. http://www.bmfiddle.com. [Online; accessed April 15, 2014].

Walny, J., Carpendale, S., Henry Riche, N., Venolia, G., and Fawcett, P. (2012). Visual thinking in action: Visualizations as used on whiteboards. *IEEE Transactions on Visualizations and Computer Graphics*, 18(12):2779–2788.

Agile Enterprise Architecture Management
An Analysis on the Application of Agile Principles

Matheus Hauder[1], Sascha Roth[1], Christopher Schulz[2] and Florian Matthes[1]
[1]Software Engineering for Business Information Systems (SEBIS), Technical University Munich,
Boltzmannstraße 3, 85748 Garching, Germay
[2]Syracom, Parkring 4, 85748 Garching, Germany
{matheus.hauder, roth, matthes}@tum.de, christopher.schulz@syracom.de

Keywords: Enterprise Architecture Management, Survey, Agile, Enterprise Architecture Framework.

Abstract: Enterprise Architecture (EA) management has proven to be an efficient instrument to align business and IT from a holistic perspective. Many organizations have established a permanent EA management function responsible for modeling, analyzing, and defining the current and future EA state as well as the roadmap. Similar as in software development, EA management initiatives face challenges that delay results, complicate the collaboration, and deteriorate the overall work quality. While in software development, agile principles and values reflected in tangible methods like Scrum and Extreme Programming are increasingly adopted by organizations, there is little known whether these practices have already made their way into EA management. Based on three research questions, this paper sheds light on the status-quo of agile principles applied to EA management. We present results of an online survey among 105 industry experts working for more than 10 industry sectors across 22 different countries.

1 INTRODUCTION

Globalization, frequently altering market conditions, and the pressing need to reduce operating costs force organizations to carry out complex business transformations at a regular interval. However, performed without a holistic and explicit picture of the organization, these transformations are likely to fail (Ross et al., 2006). An Enterprise Architecture (EA) serves as a common means to look at an entire organization as a whole. It captures both, business aspects (e.g., business processes, business objects) and IT aspects (e.g., interfaces, networks, devices) as well as their interrelations (Buschle et al., 2012).

Being applied by an increasing number of enterprises, the corresponding discipline EA management fosters the mutual alignment of business and IT (Weill and Ross, 2009).

EA management deals with capturing, modeling, analyzing, and defining the current, planned, and future architecture in conjunction with the roadmap leading from the as-is to the target state (The Open Group, 2011). However, EA management faces various challenges ranging for instance from the late return on investment to the delayed valuation of the disciplines by concerned stakeholders (cf. e.g. (Hauder et al., 2013), (Lucke et al., 2010) and (Lucke et al., 2012).

When looking on the domain of software development, researchers likewise to practitioners propose the adherence to so-called agile values helping to address these types of challenges (Schwaber, 2004). Key to these values are agile principles like the avoidance of waste (Gloger, 2010), an early stakeholder involvement (Beck et al., 2001), and gathering feedback at an ongoing basis (Highsmith and Cockburn, 2001).

In many cases these principles are based on lean production practices initially applied by the Japanese car manufacturer Toyota (Deming, 2000), (Holweg, 2007). As of today, the benefits of agile principles to software development are still discussed controversially (Reifer et al., 2003).

Several similarities between software development - centering rather on single systems - and EA management - focusing on the holistic management of systems of systems - can be drawn. Both disciplines have to handle frequently changing requirements while ensuring a close collaboration among the multi-disciplinary stakeholders. Focusing on the latter, researchers have already proposed to apply agile practices known from the development of software (Ambler, 2010a), (Buckl et al., 2011).

Given that in many cases EA management is initially promoted through IT (Hauder, Roth, Matthes and Schulz, 2013), EA initiatives are well aware of agile practices, e.g. Scrum, Extreme Programming (XP), Feature Driven Development (FDD), and might apply their according principles in their day-to-day business. From an empirical standpoint, we witness that EA management endeavors of our industry partners increasingly apply these agile practices. In addition, we diagnose agile prescriptions are implicitly incorporated in today's EA management frameworks, e.g., (The Open Group, 2011).

Yet, literature documenting the agile nature of EA management is scarce; in particular empirical ground is missing. Against this background, we conclude to the following research objective:

> *'Which agile principles known from the software engineering domain should be applied for the design of an organization-specific agile EA management practice?'*

The research approach and the deliverables are illustrated in Figure 1: After defining the scope and the research questions, we conducted a literature review to identify agile principles from other domains. Based on these principles we designed an online survey to evaluate their usage in EA management. In order to answer the research questions and gain a deeper understanding on the application of the principles, we correlated them with the specific EA management challenges of the organizations (Hauder et al., 2013).

The following section (Section 2) provides the findings we made when perusing EA management literature looking for agile pointers. In Section 3 we explain how agile principles and values from the software development world could be adopted by EA management. In addition to the introduction of our research hypotheses, Section 4 outlines the main setup of an online survey we conducted among 105 experts in the field. We analyze and discuss the data in Section 5 and 6 before concluding the paper with remarks on future research.

2 AGILE PRINCIPLES IN ENTERPRISE ARCHITECTURE

To identify tangible material on agile principles and values in EA management, we applied a structured approach as recommended by Webster and Watson (Webster and Watson, 2002). During April 2013 we perused different IS journals, conference proceedings, and books using the Web of Science, Google Scholar, IEEExplore, Citeseer, SpringerLink and the library of our research institution. Thereby, we carried out electronic full-text searches on the following English keywords: 'enterprise architecture management' and 'agile' as well as their German translations. After a first analysis of the overall 53 sources (title, abstract, outline) adhering to the method of hermeneutic text comprehension, the following sources have been identified as relevant given their focus on the topic.

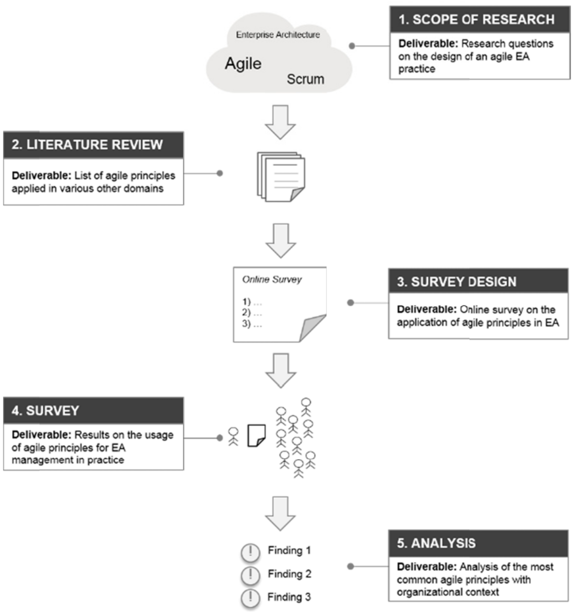

Enterprise Architecture
Agile
Scrum
1. SCOPE OF RESEARCH
Deliverable: Research questions on the design of an agile EA practice
2. LITERATURE REVIEW
Deliverable: List of agile principles applied in various other domains
Online Survey
1) ...
2) ...
3) ...
3. SURVEY DESIGN
Deliverable: Online survey on the application of agile principles in EA
4. SURVEY
Deliverable: Results on the usage of agile principles for EA management in practice
Finding 1
Finding 2
Finding 3
5. ANALYSIS
Deliverable: Analysis of the most common agile principles with organizational context

Figure 1: Research approach and deliverables.

Ambler accentuates that EA management has to be business driven, evolutionary, collaborative, and focused on producing valuable artifacts (Ambler, 2010a). Based on an examination of problems EA management is typically coping with, the practitioner points out six pieces of advice to make the management of enterprise architectures more agile, among others, simplicity, focus on people, and an iterative and incremental approach (Ambler, 2009). In the latest of his reports, Ambler goes even further, proposing an agile architecture process complemented by several key techniques (Ambler, 2010b). The findings published by Ambler all root in practical work experience with no evaluation on a quantitative scale.

Bob Rhubart describes how an EA management could be turned more agile (Rhubart, 2010). The Oracle representative highlights the necessary buy-in from architects, developers, and other stakeholders at all levels of the organization. Next to the importance of conversation in particular with the developing teams, the manager considers the involvement of enterprise architects at the project level as very crucial. Again, all suggestions are based on in-the-field work based on a single company (employee) perspective.

Friedrichsen and Schrewe see typical EA management problems (e.g. losing sight of fundamentals, becoming a slave of the EA management framework) as a reason to introduce agile values (Friedrichsen and Schrewe, 2010). The consultants advise to launch an EA management initiative with clear goals and a limited scope while always keeping potential risks in mind. In their eyes, frameworks and tools have to be considered as toolboxes that ensure to reach the stated goals more efficiently.

While Eric Landes recommends applying concrete techniques like retrospectives and lessons learned action items, iterative cycles, as well as automated acceptance criteria in the emergent design of an architecture (Landes, 2012), Scott Nelson assumes two distinct viewpoints when discussing the similarities and differences of managing enterprises architectures vs. developing software in an agile manner (Nelson, 2012).

As another industry expert and active blogger, Gabhart advises to avoid big bang EA management projects attempting to "boil the ocean", thus are too big in scope (Gabhart, 2013). Instead of that, the author proposes to start off small, building up an EA management capability in an incremental and iterative 4-step process. Lastly, the staff member Gattadahalli of the former IT Company EDS shares the knowledge of an agile management of EAs in terms of seven critical success factors (Gattadahalli, 2004).

After having introduced EA management to the reader of their book, Bente, Bombach, and Langade proposes six so-called building blocks helping to render the discipline more agile and lean (Bente et al., 2012). Benefiting from examined sources paired with their professional experience, Bente et al., describe how to streamline the architecture processes, setup an agile EA project, and foster collaboration and participation. Even though their explanations are backed by several fictitious examples, no quantitative results are provided that would prove the adoption of agile practices in EA management.

To respond to the problems often encountered in EA management, Shirazi et al. propose a framework rendering the discipline more agile (Shirazi and Rouhani, 2009). Named Agile Enterprise Architecture Framework (AEAF), the artifact consists of seven models and eleven interactions both based on agile principles and values. Even if the authors do not indicate any relations, the five viewpoints and six project aspects also included in AEAF resemble the Zachman framework (Zachman, 1987).

Although AEAF touches on several agile aspects like regular feedback or focus on cooperation, the research group's paper neither proves the empirical relevance of an agile EA management nor it validate the framework work in practice.

Rooted in lean principles, information technology architectures, and systems engineering methods, Comm and Mathaisel propose the Lean Enterprise Architecture (LEA), a three-phase structure to organize the activities for the transformation of the enterprise to agility (Comm and Mathaisel, 2010). The researchers combine their framework with concepts from the Lean Enterprise Transformation Engineering while also incorporating lean principles and practices in the resulting process. However, their work does not detail on these principles or explains how an agile enterprise should evolve its EA.

As one of the most popular approaches, The Open Group Architecture Framework (TOGAF) 9.1 does not explicitly recommend to manage an EA in an agile style (The Open Group, 2011). In turn, a more agile organization is considered as a surplus brought along by a "good" enterprise architecture. Notwithstanding, with concepts like iterations to develop a comprehensive architecture landscape and architecture, to manage changes to the organization's architecture capability, as well as appropriate stakeholder management the EA framework TOGAF promotes important agile principles.

The striving for agile principles and values enhancing the efficiency of EA management is mainly found in practitioners' circles. While only a small number of experts emphasize the misfit of both disciplines, e.g. (Nicholette, 2007), the majority of industry authors consider agile means as being well suited for EA management (Banerjee, 2011). As of January 2014, few academic publications and frameworks embrace or even mention to apply an agile management means for EAs. Studied sources are very new, indicating that the mind-set of an agile EA management is still nascent. No contribution was found that investigated on the current status quo of agile practices in industry.

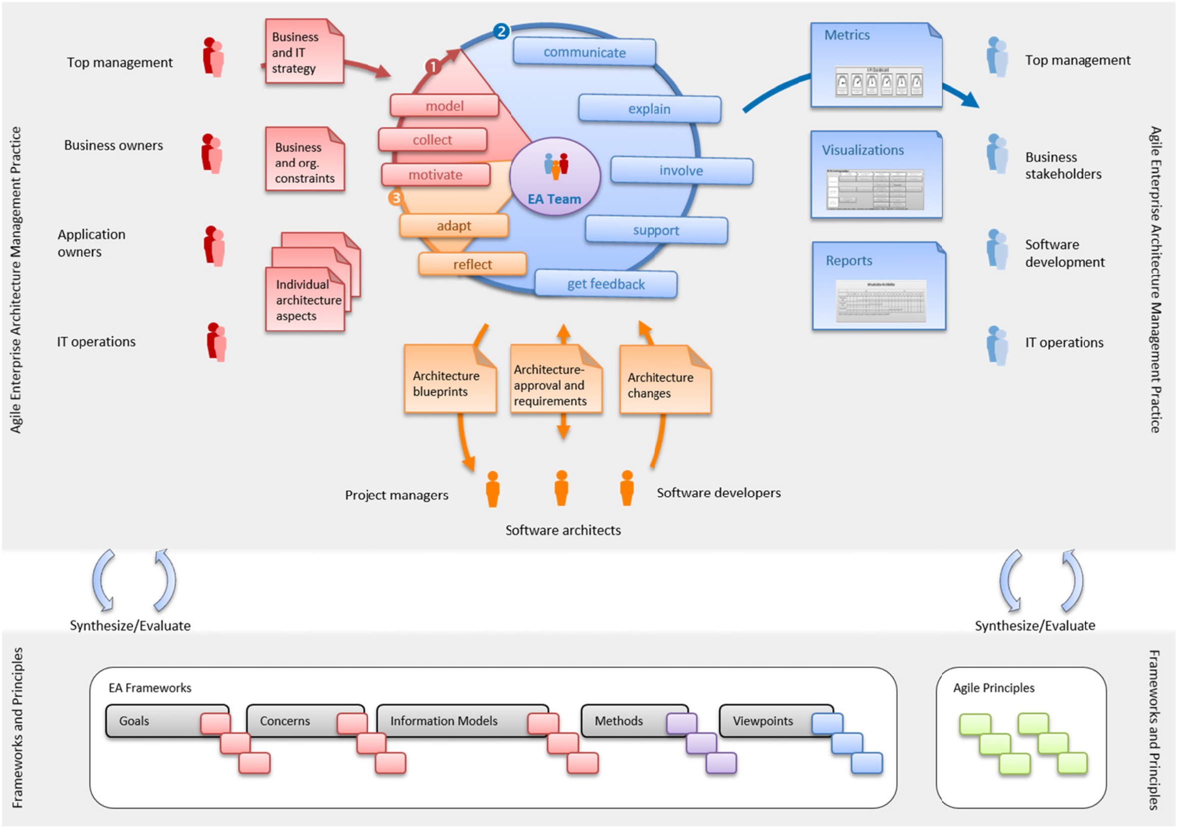


Figure 2: Design of an organization-specific agile EA management practice based on theoretical concepts often found in current EA frameworks which are extended by agile principles, cf. (Roth et al., 2014).

3 APPLYING AGILE SOFTWARE DEVELOPMENT PRINCIPLES TO ENTERPRISE ARCHITECTURE

The application of agile software development principles to EA is illustrated in Figure 2. The lower part of the figure shows the framework and agile principles that are synthesized and constantly evaluated to design an organization-specific EA management function. In the following, we illustrate the application of these principles for EA management.

When focusing on the working style applied for managing EAs, the Agile Manifesto recommends an early and constant delivery of results while maintaining a responsive attitude with regards to changing requirements (Beck et al., 2001), (Cohn, 2005). Translated into an EA management context, enterprise architects should strive to ship their deliverables as early as possible, pursue an incremental and iterative approach, and embrace changes regarding their working style and results. Similar to their software developing counterparts, an EA management team should always take care of the most important tasks first with a valuation of time over completeness and quality (Stal, 2012).

As goes the Agile Manifesto (Beck et al., 2001), an EA management team should advance in a constant pace trying to avoid overtime while having enough leeway for reflections and retrospectives. Speaking of flexibility, agile literature recommends a modus operandi where members are allowed to experiment and try out new things (Coldewey, 2012). In conforming to the pull-principle (Gloger, 2010) as well as the one-piece flow (Fisher, 2000), the EA management team should create deliverables only upon stakeholders' demand within an environment with little or no distraction and interference during the work (Schwaber, 2004). On the other side of the spectrum, stakeholders should be eager to give regular feedback (Ross, Weill and Robertson, 2006) on the results delivered through the EA management team. However, the latter should be incorporated into the work of the team.

In the sense of working software and simplicity (Beck et al., 2001), (Highsmith and Cockburn, 2001), EA management results should be as usable, simple, and accessible as possible for EA management stakeholders. Benefiting from each individual deliverable the EA management team releases, stakeholders should be satisfied with the outcome and value the EA management team creates. As called for in agile literature (Highsmith, 2002), (Gloger, 2010), EA management results should be of the highest quality, crafted in a way that they only respond to the stakeholders' demand with a level of done that is understood and agreed upon.

Centering on the actors performing the work, agile sources emphasize a cross-organizational team whose members are specialized to perform various tasks (Gloger, 2010) in a self-organized manner (Beck et al., 2001). From an educational perspective (Coldewey, 2012), the EA management team members should have special skills and training in multiple organizational areas (e.g., infrastructure, processes, application) while being capable to manage the sequence order their tasks are eventually completed.

Both, high education and expertise permit the team to speak the same language as stakeholders and information providers on a daily basis. In line with the fifth agile principles (Beck et al., 2001), the EA management team leader has to create a positive work environment while catering to the team's self-organization. Besides an intrinsic motivation (Beck et al., 2001), and work satisfaction, each EA management team member should have a notion of his/her colleagues' duties and results. Looking on the overall organizational structure (Fisher, 2000), EAM tasks should be accomplished through small sub-teams in which roles and responsibilities are clearly defined and understood. Finally, the team requires strong diplomacy and negotiation skills employed when interacting with stakeholders and EA information providers.

4 RESEARCH QUESTIONS, SURVEY DESIGN, AND EMPRICAL BASIS

In above outlined literature the application of agile principles for EA management has been widely suggested by authors in the community. However, to date neither a comprehensive list of practical applied agile principles in EA management is published nor an empirical validation thereof exists.

Since in many cases EA management is initially promoted through IT (Hauder et al., 2013) which adopts agile principles more or less eagerly, we formulate the first research question as follows:

Research Question 1: *What are frequently applied agile principles for EA management in practice?*

Our second research question aims at validating observations, e.g. on the shift towards an incremental and iterative work fashion for certain EA management challenges. Not only this enhances the scoping during the launch of EA initiatives, incremental and iterative developed products might provide stakeholders with early results and, thus, lead to an increased buy-in.

Research Question 2: *Which agile principles are used in enterprises for certain EA management challenges?*

Typically EA management puts focus on a long term plan how organization should evolve, while agile practices promote the constant change of ongoing projects. Since both approaches appear contradictory at the first look, we formulate the third research question as follows:

Research Question 3: *What are challenges for the design of an organization-specific agile EA management practice?*

To evaluate these three questions on an empirical basis, we compiled an online questionnaire using 3-point Likert scale questions. The contained questions were based on the collection of agile principles we explained above. To optimize the questionnaire's design, we conducted a pre-test with three independent non-related researchers who were requested to complete our survey.

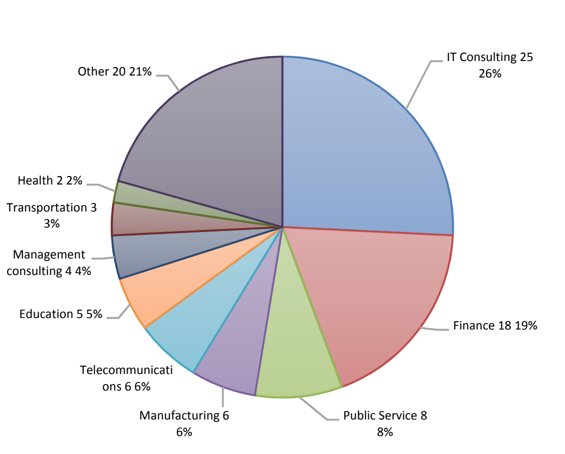


Figure 3: Industry sector of organizations (n and %).

The final version of the questionnaire was available for 21 days. To receive relevant information we targeted participants working in EA management or related fields. Using e-mail, we sent over 1100 survey invitations to industry experts we collaborated with during the last 8 years.

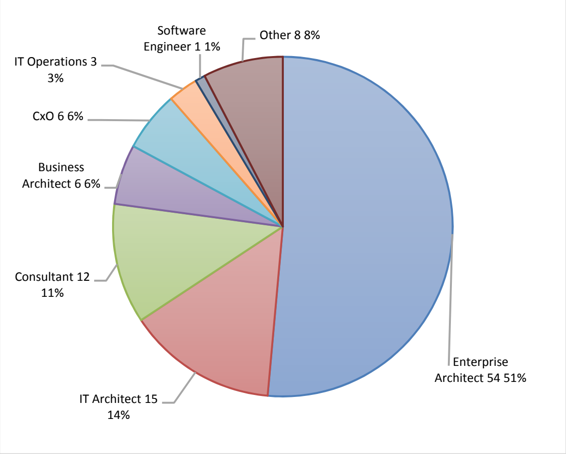


Figure 4: Job titles of participants (n and %).

In addition, the survey has been published in the two online forums Xing and LinkedIn, announcing them as topics related to EA or strategic IT management. We received input from 178 survey participants, filtered duplicate answers, and ended up with 105 completed answers for the evaluation, i.e. a dropout quote of ~41%.

As the survey was conducted primarily in Germany, 61 (~58%) participants are employed in Europe. 18 (~17%) work in the USA and 26 (~25%) are employed in other countries having less than 10 responding participants. Figure 3 illustrates the distribution of the industry sectors of the participating organizations. IT consulting is the largest sector, whereas all consultancies were requested to answer on behalf of one particular EA management engagement. IT consulting is followed by the Finance and Public sectors.

Figure 4 depicts the participants of the online survey divided by job title. The largest groups consist of Enterprise Architects followed by IT Architects and Consultants. Among the participants are also Business Architects and members of the management board. In average, questioned organizations have an experience of 5 years in EA management.

5 AGILE PRINCIPLES FOR ENTERPRISE ARCHITECTURE MANAGEMENT IN PRACTICE

In the following three subsections the research questions are evaluated based on our empirical data set. The second research question is evaluated by applying the Pearson's chi-square test to validate the dimensions in our data set.

5.1 Application of Agile Principles

The first research question deals with the application of agile principles for EA management in practice. Figure 5 illustrates the practical adoption of agile principles in EA management ordered by frequency. As depicted, organizations adhere to agile principles with a different degree of intensity, confirming our assumption that the applicability of agile principles varies for EA management. For instance, while most organizations perform retrospectives within their EA management team, only few value time over quality. Most EA management initiatives apply an iterative (~79%) and incremental (~87%) approach. About 93% of the organizations apply EA management in a self-organized manner. Moreover, ~75% say that they act cross-functionally.

While the overwhelming majority of organizations apply several agile principles for the introduction and operation of their EA management initiatives in practice, some principles are less frequently traceable. In particular some of these less frequent agile principles are related with the quality and completeness of the developed EA products.

Only ~42% of the participating organizations apply time over completeness and only ~25% rate time over quality for the developed EA products. Next to agile principles related to quality and completeness of the developed EA products, actual stakeholder demands and utilization of the produced EA products by these stakeholders are only applied by the minority of the organizations in our dataset. With ~38% only a small number organizations are truly concerned whether these EA products are actually used by stakeholders.

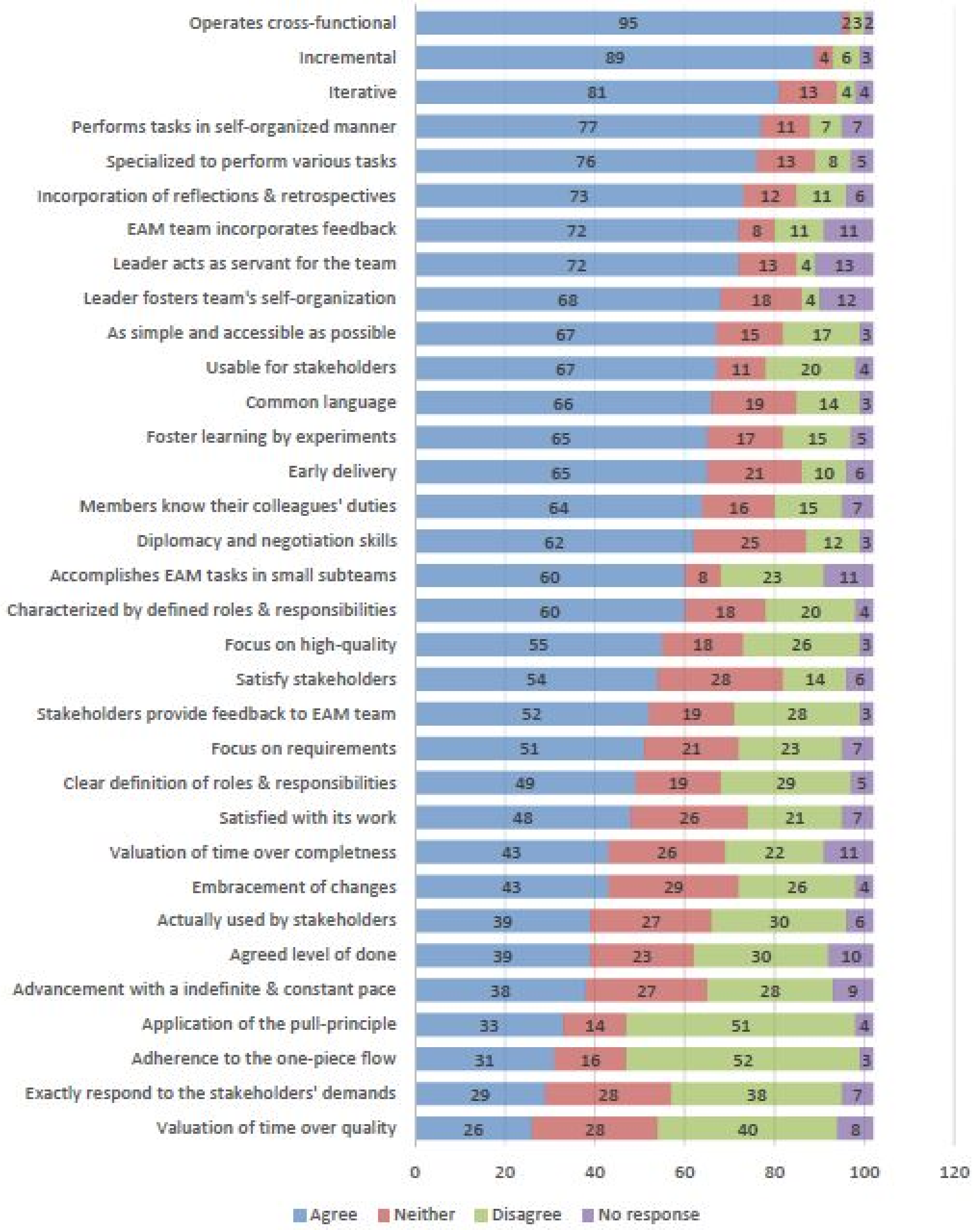


Figure 5: Applied agile principles for EA management in practice (n=105).

5.2 Agile Principles and Enterprise Architecture Challenges

We answer the second research question by correlating EA management challenges from our empirical basis (cf. Hauder et al., 2013) with the agile principles illustrated in Figure 5. Due to space limitations, we only illustrate the statistical correlations for three major EA management challenges with agile principles using Pearson’s chi-square test.

The challenge late valuation of EA management through stakeholders appears in ~51% of the participating organizations. According to our statistical test these organizations apply the principle adherence to one-piece flow with $p = .047$ ($p \leq .05$). In addition, the principle focus on requirements resulted in a goodness of fit test of $p = .00004$ ($p \leq .05$). Further agile principles that correlate with this challenge are advancement with an indefinite & constant pace p = .002 ($p \leq .05$), stakeholders provide feedback to EA management team p = .0002 ($p \leq .05$), agreed level of done p = .009 ($p \leq .05$), useable for stakeholders p = .042 ($p \leq .05$), and as simple and accessible as possible p = .005 ($p \leq .05$). All other agile principles were not statistically dependent on this challenge for the given relevance.

Around ~38% of the organizations are struggling with outdated EA results. This means that architecture descriptions are often outdated before they are complete and often understood as a project rather than a continuous process. The agile principles characterized by defined roles & responsibilities correlates with p = .004 ($p \leq .05$), members knows their colleagues' duties with p = .0001 ($p \leq .05$), focus on high quality p = .005 ($p \leq .05$), satisfied with its work p = .001 ($p \leq .05$), adherence to one-piece flow p = .00001 ($p \leq .05$), incorporation of reflections & retrospectives p = .001 ($p \leq .05$), agreed level of done p = .0001 ($p \leq .05$), and usable for stakeholders p = .001 ($p \leq .05$).

Reluctant information providers are a challenge for ~65% of the organizations. This is a very critical problem since enterprise architects heavily rely on the information and knowledge provided by stakeholders. The agile principle satisfied with its work correlates with p = .043 ($p \leq .05$), focus on requirements p = .00001 ($p \leq .05$), application of the pull principle p = .009 ($p \leq .05$), embracement of changes p = .030 ($p \leq .05$), valuation of time over quality p = .004 ($p \leq .05$), as simple and accessible as possible p = .00001 ($p \leq .05$), and exactly respond to stakeholders' demand p = .003 ($p \leq .05$) correlate with this challenge.

5.3 Designing an Agile Enterprise Architecture Management Practice

Designing an agile EA management practice is a challenging issue. While EA management frameworks typically work towards a long range vision of the organization or a business case, agile practices incorporate findings from ongoing projects immediately in the process. To put it in another way, both approaches appear contradictory due to their top-down and planning (EA management) respectively bottom-up and emergent course of action.

Regarding the challenges EA management initiatives in organizations are faced with neither of these approaches can solve all challenges on his own. Integrating both approaches within one agile EA management practice that is tailored to the specific demand of the organizational context would be desirable. The findings presented in this paper provide an initial empirical basis for further research on an agile EA management practice. This compromises the development of agile EA management roles, activities, and deliverables.

6 CONCLUSIONS

In this paper we provided an empirical foundation for agile principles applied to EA management by today's organizations. Due to the survey design, the asked industry experts could only confirm or reject the application of an agile principle for EA management. Details about their actual implementation are yet to be revealed. As of today, this might be challenging, given the scarce literature on agile EA management and only the implicit adoption through EA frameworks. Regarding our survey results, a potential bias might originate from the lack of a common understanding on how to operationalize agile principles in EA management.

Further research could examine the impact of agile principles on the success and benefits of EA management initiatives. Thereby, the efforts should account for different organizational factors like the size of the business, structure, EA management experience, industry, and tool support. Further studies could also focus on the correlation (and later causalities) between challenges encountered in EA management and possible mitigation through agile principles.

REFERENCES

Ambler, S. W. (2009): *Agile Enterprise Architecture.* http://www.agiledata.org/essays/enterpriseArchitecture.html. Last opened: 27/08/2013.

Ambler, S. W. (2010a): *Agile and Enterprise Architecture.* https://www.ibm.com/developerworks/mydeveloperworks/blogs/ambler/entry/agile_and_enterprise_architecture?lang=en. Last opened: 27/08/2013.

Ambler, S. W. (2010b): *Agile Architecture: Strategies for Scaling Agile Development*. http://www.agilemodeling.com/essays/agileArchitecture.htm. Last opened: 27/08/2013.

Banerjee, U. (2011): *Agile development and Enterprise Architecture practice - Can they coexist. Technology Trend Analysis*. http://setandbma.wordpress.com/2011/04/11/agile-development-and-enterprise-architecture-practice-can-they-coexist. Last opened: 27/08/2013.

Beck, K., Beedle, M., Bennekum, A. van, Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., et al. (2001): *Manifesto for Agile Software Development*. Agile Alliance. http://agilemanifesto.org. Last opened: 27/08/2013.

Bente, S., Bombach, U., & Langade, S. (2012): *Collaborative Enterprise Architecture: Enriching EA with Lean, Agile, and Enterprise 2.0 Practices. 1st ed*. Morgan Kaufmann, Burlington.

Buckl, S., Matthes, F., Monahov, I., Roth, S., Schulz, C., & Schweda, C. M. (2011): *Towards an Agile Design of the Enterprise Architecture Management Function*. 6th International Workshop on Trends in Enterprise Architecture Research (TEAR). Helsinki.

Buschle, M., Grunow, S., Matthes, F., Ekstedt, M., Hauder, M., & Roth, S. (2012): Automating enterprise architecture documentation using an enterprise service bus. In *Proceedings of the 18th Americas Conference on Information Systems*. Washington.

Cohn M. (2005): *Agile Estimating and Planning*, Prentice Hall PTR, Upper Saddle River.

Coldewey, J. 2012: Was heißt hier eigentlich “Agil”? Kennzeichen agiler Organisationen. In *ObjektSpektrum 05/2012*.

Comm, C. L., & Mathaisel, D. F. X. (2010): *A Lean Enterprise Architecture for Business Process Re-engineering and Re-marketing*. 12th International Conference on Enterprise Information Systems (pp. 497–500). Madeira.

Deming, W. E. (2000): *Out of the Crisis*. MIT press, Cambridge.

Fisher, K. (2000): *Leading Self-Directing Work Teams*. McGraw-Hill, New York.

Friedrichsen, U., & Schrewe, I. (2010): Leichtgewichtige Unternehmensarchitekturen – Wie Agilität bei der Einführung eines EA Management helfen kann. In *OBJEKTspektrum, EAM/2010*.

Gabhart, K. (2013): *Generating Value through Information Architecture*. http://archvalue.com/agile-enterprise-architecture (last opened: 27/08/2013)

Gattadahalli, S. (2004): *Agile Enterprise Architecture (AEA) - 7 Steps to Success*. London.

Gloger, B. (2010): Scrum. In *Informatik-Spektrum, 33(2): 195–200*.

Hauder, M., Roth, S., Matthes, F., & Schulz, C. (2013): *Organizational factors influencing enterprise architecture management challenges*. 21st European Conference on Information Systems (ECIS). Utrecht.

Highsmith J. (2002): *Agile Software Development Ecosystems*. Pearson Education, Indianapolis.

Highsmith, J., & Cockburn, A. (2001): Agile software development: the business of innovation. In *Computer, 34(9): 120–127*.

Holweg, Matthias (2007): The genealogy of lean production. In *Journal of Operations Management 25 (2): 420–437*.

Landes, E. (2012): *Agile Software Development Concepts for Enterprise Architects*. http://www.devx.com/architect/Article/47842. Last opened: 27/08/2013.

Lucke, C., Krell, S., & Lechner, U. (2010): Critical Issues in Enterprise Architecting - A Literature Review. In *AMCIS 2010 Proceedings (pp. 1-11)*. Association for Information Systems.

Lucke, C., Bürger, M., Diefenbach, T., Freter, J., & Lechner, U. (2012): Categories of Enterprise Architecting Issues - An Empirical Investigation based on Expert Interviews. In *D. C. Mattfeld & S. Robra-Bissantz (Eds.), Multikonferenz Wirtschaftsinformatik (pp. 999-1010)*. GITO mbH Verlag, Berlin.

Nelson, S. (2012): *Making Enterprise Architecture Work in Agile Environments*. http://www.devx.com/architect/Article/47871. Last opened: 27/08/2013.

Nicholette, D. (2007): *Enterprise Architecture and Agile. Musings of a Software Development Manager*. http://edgibbs.com/2007/10/04/enterprise-architecture-and-agile. Last opened: 27/08/2013.

Reifer, D. J., Maurer, F., & Erdogmus, H. (2003): *Scaling agile methods. Software, IEEE, 20(4): 12-14.*

Roth, S., Zec, M. & Matthes, F. (2014): *Enterprise Architecture Visualization Tool Survey 2014*. Technical Report, Technische Universität München.

Ross, J. W., Weill, P., & Robertson, D. (2006): *Enterprise architecture as strategy: Creating a foundation for business execution*. Harvard Business Press, Boston.

Webster, J., & Watson, R. T. (2002): *Analyzing the Past to Prepare for the Future: Writing a Literature Review.* MIS Quarterly, 26(2): 13–23.

Weill, P., & Ross, J. W. (2009): *IT Savvy: What top executives must know to go from pain to gain*. Harvard Business Press, Boston.

Rhubart, B. (2010): *Agile Enterprise Architecture*. Oracle Magazine. http://www.oracle.com/technetwork/issue-archive/2010/10-nov/o60architect-175580.html. Last opened: 27/08/2013.

Schwaber, K. (2004): *Agile Project Management with Scrum. 1st ed.*, Microsoft Press, Redmond, Washington.

Shirazi, H. M., Rouhani, B. D., & Shirazi, M. M. (2009): A Framework for Agile Enterprise Architecture. In *International Journal of Intelligent Information Technology Application, 2(4): 182–186.*

Stal, M. (2012): *Softwarearchitektur und Agilität - Freund oder Feind?*. OOP 2012. München.

The Open Group (2011): *TOGAF® Version 9.1*, Van Haren Publishing, Zaltbommel.

Zachman, J. A. (1987): A framework for information systems architecture. In *IBM systems journal, 26(3): 276-292.*

Context-Sensitive Impact Analysis for Enterprise Architecture Management

Melanie Langermeier, Christian Saad and Bernhard Bauer
Software Methodologies for Distributed Systems, University of Augsburg, Germany
{langermeier, saad, bauer}@ds-lab.org

Keywords: Enterprise Architecture Analysis, Impact Analysis, Change Propagation, Data Flow Analysis.

Abstract: Since Enterprise Architecture (EA) models are typically very large, it is often difficult for humans to fully grasp their contents. Due to this inherent complexity, the task of generating additional value from these models is very challenging without a suitable analysis method. Impact analysis, which is able to determine the effects which changes have on other architectural elements, can therefore provide valuable information for an enterprise architect. Whether an element is affected by a change depends on its context, i.e. its (transitive) connections to other elements and their status with respect to the analysis. In this paper we propose a context-sensitive approach to the implementation of impact analyses. This method relies on the technique of data-flow analysis to propagate the effects of changes throughout the model. As a consequence, the specification can be defined in a very generic fashion, which only relies on relationship classes. Therefore it can be easily adapted to organization-specific EA meta models as only the relationship types have to be mapped to the respective classes.

1 INTRODUCTION

Enterprise Architecture Management (EAM) provides methods for managing the inherent complexity of the large IT infrastructures encountered in many organizations. As a result, Enterprise Architecture (EA) models usually contain many elements which are connected through complex relationships. It is therefore vital to provide suitable methods for (semi-)automatically analyzing their contents to be able to benefit from this methodology once it has been successfully established in an organization.

Although much research has been done in the EA domain, most of this work focuses on methodologies for the development and the representation of enterprise models. By contrast, approaches and techniques which explore possible applications scenarios are very rare (Närman et al., 2012; Niemann, 2006). Regarding the analysis of EA models, a major focal point exists in their quantification. This encompasses the definition and computation of quality attributes such as application usage and service availability. (Närman et al., 2012) Furthermore, it is possible to evaluate the performance and cost aspects in the different layers of enterprise models (Jonkers and Iacob, 2009). Finally, (Matthes et al., 2012) establish a catalog of KPIs to measure EA management goals.

One of the most important analysis methods however, is the so-called impact analysis which allows to simulate the effects of changes (e.g. the modification of a CRM system) and to assess risks in the current architecture (e.g. which business operations would be affected if a specific server goes offline) (de Boer et al., 2005). To generate this information, an impact analysis has to evaluate the dependencies between the architecture's constituents. However, in order to make proper assertions about these relationships, it is necessary to evaluate each element in its respective context. This means, that its relationships with other elements in the model have to be taken into consideration. For example, to examine the impact of a server failure on business processes, one has to determine which applications rely on this server. This requires a careful evaluation of indirect and transitive paths in the model to ensure that all necessary information is retrieved, while at the same time excluding irrelevant relationships.

Existing approaches and tools for the creation and analysis of EA models usually rely on a static meta model structure. This can be a problem since each organization tends to employ its own meta model, making the adaption of existing analyses very difficult (Kurpjuweit and Aier, 2009). To rectify this situation, more flexible methods for handling structural dependencies are required.

In this paper we present a technique which supports the context-sensitive impact analysis of EA models. It is based on the principle of data-flow analysis, a method which originates from the field of compiler construction. Using this approach, it is possible to derive context-sensitive information by propagating contextual information along the model's edges. Since the developed analysis distinguishes between different semantic relationship classes it can be easily adapted to the conventions in different organizations by mapping the relationship types in the respective target domain to the proposed categories. Furthermore, it is possible to extend the analysis with individual impact propagation rules. To demonstrate the viability as well as the generic applicability of this approach, we implement multiple impact analyses for different EAM languages.

2 IMPACT AND DEPENDENCY ANALYSIS

According to (Bohner, 2002), determining the effects of a change requires an iterative and discovery-based approach. Change impact analysis can be performed for a single software system, but also on an architectural level for a full application landscape or an enterprise architecture. A related topic which is also of interest in this context is the analysis of dependency relationships.

Typically, any change which is made to a model element also affects its neighboring elements (*direct impact*). However, as these changes may in turn affect other elements (*indirect impact*), the effect propagates throughout the model. Consequently, even a small change in a single element can cause ripple-effects, resulting in non-trivial consequences. While the direct impact can be derived from the connectivity graph, the computation of indirect impacts (*n-level impacts*) requires reachability information. However, since this method approximates potential impacts, it tends to overestimate the result by generating false-positives. The precision of the analysis can be improved by using a constraint mechanism or by incorporating structural and semantic information (Bohner, 2002).

Most of the work regarding impact analysis of software focuses on the code level (Lehnert, 2011). Approaches which evaluate architectures usually only regard concepts such as components, packages, classes, interfaces and methods. Due to the limited amount of supported types and the domain-specific characteristics, these approaches are not suitable for use in EAM.

Nevertheless, some techniques which target the UML are more closely related to the EAM domain. (Briand et al., 2003) propose a methodology for subjecting analysis and design documents to an impact analysis to detect side effects of changes in the context of UML-based development. To restrict the set of affected model elements they propose the use of a coupling measure and a predictive statistical model. The impact analysis itself is specified using the OCL. (von Knethen and Grund, 2003) developed an approach which supports traceability by providing requirements engineers, project planers and maintainers with the ability to monitor the effects that changes have on software systems. They differentiate between three types of relationships to define the traces: *representation*, *refinement* and *dependency*. To determine the change impact, they (semi-)automatically analyze requirement traces using these three categories.

(Kurpjuweit and Aier, 2009) and (Saat, 2010) propose techniques for EA dependency analysis. Saat focuses on time-related aspects (org. "zeitbezogene Abhängingkeitsanalysen") by considering for each element its life time, the status (current or proposed) as well as the life cycle phase with its duration. However, no execution or implementation details are provided for this approach. Kurpjuweit and Aier developed a formal method for flexible and generic dependency analysis. To determine dependent elements, they use the transitive closure of a set of relations. They also define an expansion function, which allows to consider special relation semantics, e.g. hierarchical refinement or reflective relation types.

(Holschke et al., 2009) as well as (Tang et al., 2007) propose the use of Bayesian Belief Networks (BBN) for EA modeling. These approaches rely on causal dependencies as well as inference methods for BBN and a diagnosis analysis to determine the impact. The former realizes a failure impact analysis, theoretically described in the pattern catalogue (Buckl et al., 2008), using the diagnostic analysis[1] and the modeling tool GeNIe. As a result, architectural components can be ranked with respect to their criticality for a business process. However, this approach focuses on availability, not on changes. Tang et al. employ a combination of predictive reasoning to determine affected elements and diagnostic reasoning to determine the cause of a change. Prior to the analysis, the architect has to assign a probability to each root node and a conditional probability table to each

[1] Jagt, R.M.: Support for Multiple Cause Diagnosis with Bayesian Networks. Vol. M. Sc. Delft University of Technology, the Netherlands and Information Sciences Department, University of Pittsburgh, PA, USA, Pittsburgh (2002)

non-root node.

Propagation rules are another method for determining the impact of changes. This technique allows to define effects that depend on structural and semantical properties. An iterative application of those rules to a model yields the direct and indirect impacts. (de Boer et al., 2005) present such rules for the most important relationships in ArchiMate models, differentiating between the *removal*, the *extension* and the *modification* of an architectural element. However, the definitions are given in an informal and textual manner and no technical realization is supplied. (Kumar et al., 2008) propose rules that encode the dependency relationships of the attributes of entities. Changes are thereby propagated to determine the impact on a defined set of element types, namely business goals, processes, services and infrastructure components as well as the relations *runs on*, *provides*, *executes* and *delivers*. No mechanism is specified for implementing the change propagation. (Aryani et al., 2010) also rely on the propagation concept to define a conceptual coupling measurement for software components. Based on this information a dependency matrix is established which allows to predict change impacts.

In (Lankhorst, 2012), a tool for impact-of-change analysis is described. The author represents enterprise architectures in XML and uses the Rule Markup Language (RML) to define transformations which represent the rules which define the impact-of-change. The RML rules are analyzed through a pattern matching of the antecedent against the input XML. If a rule matches, the variables will be bound and an output XML is generated based on the rule output.

3 A CONTEXT-SENSITIVE APPROACH TO IMPACT ANALYSIS

The foundation for the definition of any impact is the computation of reachable elements. According to (Bohner, 2002), *reachability* denotes transitive connections, whereas *dependability* refers to directly connected elements. To determine reachability relationships we employ data-flow analysis, a technique which is based on the principle of information propagation. This allows to directly implement the following recursive specification: An element is reachable if at least one predecessor element is reachable. In this context, a predecessor is defined as the source element of an incoming edge. Since there are typically no isolated areas in an EA model, this would normally result in almost all elements being classified as reachable. For a more focused analysis, we therefore need to extend the reachability computation with contextual information. For this purpose, we establish two different categorization mechanisms for relationships. For each relationship class in these categories we define a change propagation rule which specifies how a change will be propagated through the model.

In the following we will first formalize the representations of model and meta model data in a way which ensures the applicability of the approach even if an organization employs a customized version of the meta model. We will then describe a data-flow based specification of a naive reachability analysis and subsequently propose extensions which enable a context-sensitive analysis of change impacts.

3.1 Formalizing the Meta Model and the Model

The high diversity of meta models results in a major challenge when devising techniques in the context of EAM. To overcome this issue, we developed a generic meta model which is able to support any EA language based on traditional modeling paradigms. Apart from abstracting from the particular structure of an input language, this approach has the benefit of combining meta model and model data in a single representation.

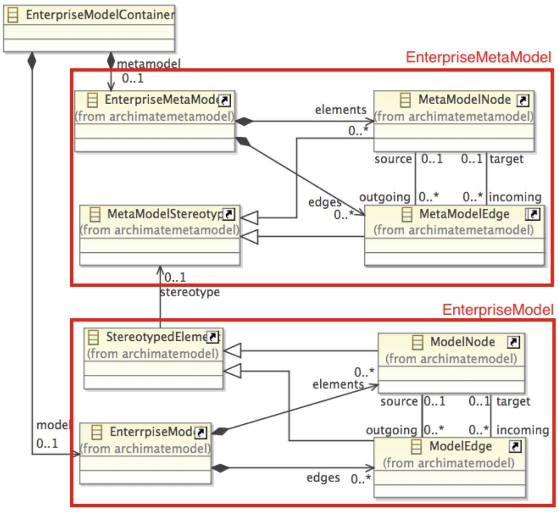


Figure 1: Generic representation for EA (meta) model data.

A condensed version of this specification is depicted in figure 1. The relevant elements can be described as follows: Each concept of the respective target EA language is translated into either a *MetaModelNode* or a *MetaModelEdge*. Connections between these elements have to be established accordingly during the transformation process. Both types also carry additional meta information such as their stereotype, the concept's name and its properties. Instances from the target EA model are converted into *ModelNodes* and *ModelEdges* and connected to their respective meta model stereotypes.

3.2 Analyzing Reachability for EA Models

The computation of reachability information forms the basis for the subsequent impact analysis. An element is declared reachable, if there exists a path connecting the element to the starting point (*indirectly connected elements*). The reachability analysis is carried out using the Model Analysis Framework (MAF) (Saad and Bauer, 2011) which supports the specification and execution of data-flow based analyses on models.

Data-flow analysis is used by compilers to derive optimizations by examining the structural composition of program instructions. Canonical examples include the calculation of reaching definitions and variable liveness. For this purpose, the program is converted into a control-flow graph with the nodes representing the basic blocks and the edges denoting the flow of control. A set of data-flow equations is then evaluated in the context of each node. Each equation takes the results computed at the immediate predecessor nodes as input, applies a confluence operator (union or intersection) to combine these sets and finally modifies the values to reflect the effects of the local node's instructions. Effectively, this method describes an equation system which propagates information throughout the underlying graph, thus enabling a context-sensitive evaluation of each instruction. If loops are present, fixed-point evaluation semantics are employed to approximate the runtime behavior of the program.

In (Saad and Bauer, 2013) we discussed an adaption of this analysis technique to the modeling domain which we referred to as a *generic "programming language" for context-sensitive model analysis*. This approach defines a declarative specification language that allows to annotate data-flow attributes at meta model classes that can subsequently be instantiated and evaluated for arbitrary models. This technique has several significant advantages: Data-flow analysis provides inherent support for the implementation of recursive specifications which iteratively propagate information throughout a model. Also, since information is routed along model edges, each model element can be evaluated in its overall context, thus eliminating the need for static navigational expressions which are common in languages such as OCL. This is important in the EAM domain where the structure of both meta models and models is highly dynamic. Finally, the usage of fixed-point semantics allows to implement a correct handling of cyclic paths.

Using MAF, a reachability analysis for model elements can be specified in the following way:

```
 1: analysis reachability_analysis {
 2:   attribute is_reachable : Boolean initWith false;
 3:   extend node with {
 4:     occurrenceOf is_reachable calculateWith
 5:       self.incoming.source.is_reachable()
 6:         ->includes(true);
 7:   }
 8:   extend startnode with {
 9:     occurrenceOf is_reachable calculateWith true;
10:   }
11: }
```

As described above, an element $e1$ is reachable from another element $e2$, if there exists a path between $e1$ and $e2$. Here, we assume that the meta model defines the classes *node* and *startnode*, the latter one being a specialization of the former one. We further classify changed elements in the model as *startnodes* for the analysis. The reachability status is computed by a data-flow attribute *is_reachable* of type *boolean* which is initialized with the value *false* (line 2). Lines 3-7 attach this attribute to all instances of the *node* class. To determine the reachability status of a node, the data-flow equation in lines 5-6 accesses the *is_reachable* values computed at the respective node's predecessors, thereby directly implementing the recursive specification. Finally, lines 8-10 overwrite this equation at *startnodes* which are, by definition, always reachable.

3.3 Context-aware Change Propagation

The execution of the reachability analysis in section 3.2 will result in an large result set, containing mostly false positives regarding change impact. By enriching the rules with context-specific declarations, the impact set can be restricted to contain only meaningful data and to additionally reflect different types of changes.

In the following, we will differentiate between the change types *extend*, *modify* and *delete* as proposed by (de Boer et al., 2005). Extensions refer to cases where new issues are added but the initial functionality or structure remains the same. Consequently, extensions do not propagate to depending elements. By contrast, a modification also affects the functionality or the structure and therefore it cannot be guaranteed that initially provided issues will still be available or that their behavior remains unchanged. Finally, deletion indicates that an element will be removed from the enterprise architecture. The change types are prioritized as follows: *delete* overrides *modifies* overrides *extends* overrides *no change* (NO). Depending on the respective requirements, additional change types can be implemented.

Due to the lack of detailed information in enterprise architecture models, an accurate definition of the impact of a change is not possible. We therefore propose to approximate the impact using a worst case and a best case analysis similar to the practices in software analysis. For the worst case, the impact is defined as the maximal set of affected elements, whereas the best case includes only the minimal set. The real impact typically lies somewhere between both cases.

To implement the context-dependent impact analysis, we define rules which are able to differentiate between the different change and relationship types. To make the technique generically applicable, we introduce custom relationship classes to which the specific relationship types in the target EA language can be mapped. The developed specifications can be divided into two categories: In section 3.3.1 describe how the propagation of effects is influenced by different relationship classes while section 3.3.2 introduces an additional classification along different effect types.

3.3.1 Change Propagation Rules Depending on Relationship Classes

To classify the relationships of an enterprise architecture, we grouped them according to their semantics, which we identified through a literature review of existing EA frameworks and their meta models. This includes the Core Concepts Model (CC) of ArchiMate (The Open Group, 2012) and the DM2 Conceptual Data Model of DoDAF (U.S. Department of Defense, 2010).

Overall, we were able to identify five classes of relevant EA relationship types: *Locate* denotes the allocation to some location or organization unit. Any kind of provision of functionality, information and behavior is of the type *provide* while the *consume* class denotes the consumption of those elements. *Structural dependency* relationships define the structure or organization of entities in one layer. The *behavioral dependency* class on the other hand summarizes relationships which declare dependencies between the behavior of elements in a single layer which are neither of the type provide nor consume. The following table lists all classes along with corresponding examples from the ArchiMate Core Concepts and the DoDAF DM2.

Table 1: Classification of EA relationships.

class	**examples**
`locate`	*CC*: assignment *DM2*: is-at
`provide`	*CC*: realize, assess *DM2*: provide, performedby
`consume`	*CC*: used by, access *DM2*: consume
`structural dependency`	*CC*: aggregate, composite *DM2*: part-of
`behavioral dependency`	*CC*: trigger, flow to

Note that the mapping in table 1 is only a suggestion based on our interpretation of the concepts and has to be adapted if an organization assigns different semantics to these types. It is also important to realize that each relationship may belong to multiple categories. In the worst case analysis, the strongest rule will be chosen while the best case analysis will use the weakest one.

To formalize the change semantics of these classes, we employ the following syntax:

$$A.X \rightarrow B.Y \tag{1}$$

This statement indicates that if element A is changed in the manner X then element B has to be changed in manner Y. A and B represent the source and the target of the relationship while X, $Y \in \{$*modify*, *delete*, *extend*$\}$. It is also possible to cluster change operations on the left hand side. $A.\{X,Y\} \rightarrow B.Z$ means that if A is changed in the manner X or in the manner Y, B has to be changed in the manner Z. Optionally, it is possible to differentiate between a worst case (WC) and a best case (BC) impact on the right hand side of the rule.

We will now demonstrate this concept using the location relationship. Assuming that an application component (A) is hosted by a organization unit (B), this connection is mapped to the class *located at*. If a change to the application component has no effect on the organization unit the rule will be $A.\{del,mod,ext\} \rightarrow B.NO$. If, on the other hand, the organization unit is deleted, the application component loses its host. In the worst case it needs to be deleted as well while in the best case it will simply be assigned to another host. This is formalized as: $B.del \rightarrow WC:A.del,\ BC:A.ext$. Finally, if the organization unit is modified or extended, the worst case demands that the application component has to be modified too while, in the best case, it remains as is. This can be addressed with the rule: $B.\{ext,mod\} \rightarrow WC:A.mod,\ BC:A.NO$. Change rules for other relationship classes are defined in a similar manner as shown in table 2.

Table 2: Impact rules for the relationship classes.

class	rule
`located at`	A.{del,mod,ext} → B.NO B.del → WC: A.del BC: A.ext B.{ext,mod} → WC: A.mod BC: A.NO
`provides`	A.del → WC: B.del BC: B.ext A.mod → WC: B.mod BC: B.NO A.ext → WC: B.ext BC: B.NO B.{del,mod,ext} → A.NO
`consumes`	A.{del,mod,ext} → B.NO B.{del,mod} → WC: A.mod BC: A.ext B.ext → A.NO
`structurally dependent`	A.del → WC: B.del BC: B.mod A.{mod,ext} → B.NO B.{del,mod} → WC: A.mod BC: A.NO B.ext → WC: A.ext BC A.NO
`behaviorally dependent`	A.{del,mod,ext} → B.NO B.{del,mod,ext} → A.NO

3.3.2 Change Propagation Rules Depending on Effect Types

In addition to the classification along the lines of relationship types, a differentiation between different effect types can be useful as well. We therefore define the following three effects: *strong*, *weak* and *no effect*. The type of effect has to be specified for each direction of a relationship. The notation $X-Y$ indicates that a change in the source has a effect of type X on the target and vice versa. Overall, this leads to six effect classes: *Strong-Strong*, *Strong-Weak*, *Strong-No effect*, *Weak-Weak*, *Weak-No effect* and *No effect-No effect*.

The semantics of these effects can be defined using rules similar in nature to those presented in section 3.3.1. They are shown in table 3.

Table 3: Impact rules for the effect classes.

effect	rule
`strong`	A.del → WC: B.del, BC: B.ext A.mod → B.mod A.ext → B.ext
`weak`	A.del → WC: B.mod, BC: B.no A.mod → WC: B.mod,BC: B.ext A.ext → WC: B.ext, BC: B.NO
`no effect`	A.{del,mod,ext} → B.NO

If *A* strongly affects *B*, this indicates that if *A* is deleted, in the worst case, *B* has to be deleted as well and, in the best case, it only needs to be extended. A modification in *A* leads to a modification of *B* and the same applies to extensions. If, for example, an application component realizes a service, then the application component has a strong impact on the service while the service may only have a weak impact on the application component. This specific interpretation of *realize* would result in an assignment to the *Strong-Weak* class. A weak effect denotes that the deletion of *A* conducts no change in *B* in the best case and a modification in the worst case. A modification of A in the worst case requires a modification of *B*. In the best case it has only to be extended. Finally if A is extended, in the best case B must not be changed, in the worst case it has to be extended, too. If the relationship is mapped to *no effect*, any change of *A* has no effect on *B*.

Further examples for effect mappings of ArchiMate relationships are:

- Strong-Weak: realize
- Strong-No effect: aggregation
- Weak-No effect: use, assign
- Weak-Weak: triggers

3.3.3 Realization of the Rules

The rules defined in sections 3.3.2 and 3.3.1 can be realized as data-flow equations. First, the meta model and model data has to be converted to the generic representation presented in section 3.1. Then, the status of the changed elements whose impact should be analyzed is set to the respective value while the result for all other elements is initialized with *no change*. Afterwards, these values will be iteratively recomputed to propagate the effects of the changes. For illustration purposes, we include a Java-based implementation of the rule which calculates the best case result based on the presented effect types:

```
Object node_changestatus_bestcase(Node currentNode){
  for (Edge edge : currentNode.getIncomingEdges()){
    Status sourceStatus = edge.source.getStatus()
    Status currentStatus = currentNode.getStatus()
    if (edge.effectClass == StrongEffectTarget)
      if (sourceStatus == (DEL||EXT))
        return computeStatus(currentStatus, EXT)
      else if (sourceStatus == MOD)
        return computeStatus(currentStatus, MOD)
    if (edge.effectClass == WeakEffectTarget)
      if (sourceStatus == (DEL||EXT))
        return computeStatus(currentStatus, NO)
      else if (sourceStatus == MOD)
        return computeStatus(currentStatus, EXT)
    if (edge.effectClass == NoEffectTarget)
      if (sourceStatus == (DEL||MOD||EXT))
        return computeStatus(currentStatus, NO)
  }
  for (Edge edge : currentNode.getOutgoingEdges()){
    ...
  }
}
```

The status of the current element (*currentNode*) depends on the status of the connected elements as well as the direction of the relationship. Therefore, to correctly determine the change status, all incoming (lines 2 - 18) and outgoing edges (lines 19 - 21) have to be processed. The status value which has been computed for a connected element is retrieved through an invocation of *getStatus()* (line 3). This call instructs the data-flow solver to recursively compute and return the requested value. Based on the type of each incoming edge, it is then decided whether it has a *strong effect* (line 5), a *weak effect* (10) or no effect (15) on its target. The concrete type of the change is determined by evaluating the status of the edge's source element (lines 6-9, 11-14, 16-17). Finally, *computeStatus()* is invoked to compute and return the status of the local element. To implement the priorization relationships between the change types, e.g. to ensure that a weak change like *no change* cannot override a stronger one like *delete*, this method takes both the current and the newly computed status as input. A similar approach is used to calculate the result for the source elements of outgoing edges (line 19-21).

3.4 Customization of the Impact Analysis

In the case where the rules proposed in section 3.3 are not sufficient to capture all requirements of the organization, it is possible to customize the analysis. For example, if a specific relationship type cannot be mapped to one of the proposed classes, a new rule can be created. In addition to evaluating relationship types and change status of connected elements, a rule may also consider the type of the connected elements or class properties. It would also be possible to extend the rule definitions with the ability to quantify a change (e.g. in terms of costs). These features can be implemented through additional data-flow attributes. For example, to compute potential savings on IT maintenance, the maintenance costs of all deleted application and infrastructure components and their corresponding services could be aggregated.

Another customization consists of a modification of the rule set to support change probabilities. Instead of a single status, we can define four separate data-flow attributes, which compute the respective probabilities for the types *delete*, *modification*, *extension* and *no change*. Additionally, the rule specification would have to be extended. For example, a rule could be defined as:

$$P(A.del) = X \rightarrow P(B.del) = 0.8 \times X \qquad (2)$$

This means that if the probability that A is deleted is X, then the probability that B has to be deleted is $0.8 \times X$ or, in other words, if A is deleted then in 80% of the cases B will be deleted as well.

4 EVALUATION

Most of the research work regarding change impact analysis has been carried out theoretically and thus has not yet been applied to real architecture models (e.g. (de Boer et al., 2005), (Kurpjuweit and Aier, 2009), (Kumar et al., 2008)). An exception exists in the work of (Tang et al., 2007) who employ predictive and diagnostic reasoning in BBN. However, one disadvantage of their approach can be found in the high effort required to annotate probability information. The technique proposed in this paper simplifies analysis specification through a generic representation of model data and through predefined and extensible categorizations of relationships and effects.

Many existing approaches do not address problems relating to cyclic dependencies or contradicting results, the latter one for example being a weakness of the tooling proposed by (Lankhorst, 2012). Furthermore, the issue of the scalability of the technique which is based on pattern matching and model transformations is not considered and the employed RML technique is highly dependent on specific usage scenarios as well as on the respectively chosen EA language. By utilizing the data-flow analysis method with its inherent support for cyclic dependencies, recursive specifications and iterative result computation, we are able to address these challenges. The scalability of DFA (and the Model Analysis Framework in particular) has been demonstrated in the context of other domains including the analysis of extensive AUTOSAR models (Kienberger et al., 2014).

For a practical evaluation, we implemented the proposed methods in the form of an addin for MID Innovator (MID GmbH, 2014) using the MIDWagen example which is shipped with the tooling. MIDWagen describes the IT landscape of a car rental organization with its actors, business services, business processes, application components and services as well as the required infrastructure components and services. Although it is not a real world example, the level of detail and the extensibility of the underlying EA language enables a thorough evaluation of the viability and the robustness of our technique.

To illustrate the application of our approach, we focus on a modification of the *Booking System*. We assume that this component, which is responsible for payment transactions, has to be modified due to security issues. We further state that this change will only affect the *Payment* service while the *Bonus Booking* service does not have to be adapted. The modification of the *Payment* application service (AS) also causes a change to the supporting *Return* service, which in turn will affect the *Payment* business service (BS) as well as the *Renter* role. Figure 2 shows the respective excerpt from the MIDWagen model.

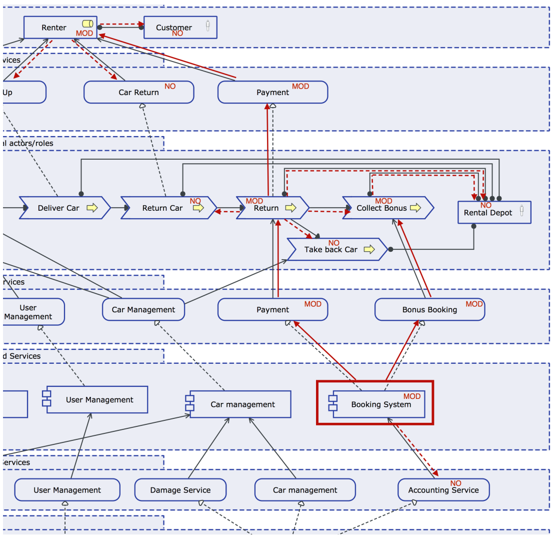


Figure 2: Excerpt of the ArchiMate model for the MIDWagen Use Case (MID GmbH, 2014) with the worst case change propagation path.

For this scenario, we employ the relationship classification described in section 3.3.1. Since the model is given in the ArchiMate language, we are able to use the mappings listed in table 1.

In figure 2, the resulting worst case change propagation path is indicated by red arcs. Solid lines represent paths along with a change is forwarded, while dashed lines stand for relationships, which are considered but do not result in a change propagation. The final impact set consists of the elements {*Payment AS* modified, *Bonus Booking* modified, *Return* modified, *Collect Bonus* modified, *Payment BS* modified, *Renter* modified}, while the final result set for the best case analysis is empty. Both results represent realistic approximations which have to be interpreted considering the severity of the modification. In the best case scenario (e.g. performance issues), the modification of the *Booking System* does not affect the provided functionality, and therefore the service does not need to be changed. In the worst case, for example a substantial change in the functionality due to security issues, the effects of the change propagate to the role in the business layer which is potentially affected by the modification. Both impact sets represent approximations, which can be of great value for estimating the real effects of a change especially in early design stages.

Carrying out the analysis using the effect classes, the result consists of {*Payment AS* modified, *Return* modified, *Payment BS* modified, *Renter* modified}. This result, which lies in between the worst and the best case of the relationship analysis, is able to provide more detailed information in the case where the required data is available.

5 CONCLUSIONS

In this paper we proposed a context-sensitive impact analysis technique for EA models. The approach relies on two underlying concepts: The problem of diverse EA languages is addressed by a generic representation of model data while the data-flow analysis method enables an intuitive specification of analyses, which depend on the iterative propagation of results.

We argued that a traditional reachability analysis which returns all direct and indirect neighbors of an element is not suitable in the EA context and therefore has to be extended with context-sensitive propagation rules. For this purpose, we defined a relationship classification which reflects the semantics of different edge types as well as the semantics of the change types *extend*, *modify* and *delete*. By applying this concept, change effect propagation depends both on the change type as well as on the meaning of the relationships which connect the respective elements. For example, if an element which is still in use is deleted, this change will affects consumers. If, on the other hand, the element is extended, leaving the existing functionality unchanged, the potential effect is not propagated.

For a more focused analysis, we proposed two different kinds of relationship classifications. Section 3.3.1 categorizes relationships according to their semantics with respect to the architecture by defining the classes *located at*, *provides*, *consumes*, *structurally dependent* and *behaviorally dependent*. In section 3.3.2, the severity of a change on the respective source and the target elements is considered by introducing the categories *strong*, *weak* and *no effect*. In both cases we defined propagation rules in the form of DFA equations for best and worst case analysis. By extending these definitions, it is possible to include support for organization-specific semantics and additional relationship types. Executing the analysis using the DFA solver of the MAF framework yields the results which can be interpreted as estimations that reflect the best and worst case of the actual impact.

The combination of the generic model representations, extensible DFA-based analysis specifications and the classification approach for relationships ensures that this technique can be applied to the various EA conventions found in different organizations. Further work has to be done to determine a suitable visualization of the results. It would also be interesting to evaluate rules for other impact scenarios such as failure impact analysis which analyzes the availability of architecture elements. Finally, it should be explored how the computation of best and worst case results could be improved through an integration of probability distributions. At the moment we only support a simple and naive way for the integration of probabilities.

ACKNOWLEDGEMENTS

This work was partially sponsered by the FuE-Programm Informations- und Kommunikationstechnik Bayern. The authors would like to thank MID GmbH for providing their demo use case, licenses for their tool as well as for their support during the implementation.

REFERENCES

Aryani, A., Peake, I., and Hamilton, M. (2010). Domain-based change propagation analysis: An enterprise system case study. In *2010 IEEE International Conference on Software Maintenance (ICSM)*, pages 1–9.

Bohner, S. (2002). Software change impacts-an evolving perspective. In *International Conference on Software Maintenance, 2002. Proceedings*, pages 263–272.

Briand, L., Labiche, Y., and O'Sullivan, L. (2003). Impact analysis and change management of UML models. In *International Conference on Software Maintenance, 2003. ICSM 2003. Proceedings*, pages 256–265.

Buckl, S., Ernst, A., Lankes, J., and Matthes, F. (2008). Enterprise Architecture Management Pattern Catalog (Version 1.0). Technical Report TB 0801, Technical University Munich, Chair for Informatics 19.

de Boer, F., Bonsangue, M., Groenewegen, L., Stam, A., Stevens, S., and van der Torre, L. (2005). Change impact analysis of enterprise architectures. In *Information Reuse and Integration, Conf, 2005. IRI -2005 IEEE International Conference on.*, pages 177 – 181.

Holschke, O., Nrman, P., Flores, W., Eriksson, E., and Schnherr, M. (2009). Using enterprise architecture models and bayesian belief networks for failure impact analysis. In *Service-Oriented ComputingICSOC 2008 Workshops*, page 339350.

Jonkers, H. and Iacob, M.-E. (2009). Performance and cost analysis of service-oriented enterprise architectures. *Global Implications of Modern Enterprise Information Systems: Technologies and Applications, IGI Global*.

Kienberger, J., Minnerup, P., Kuntz, S., and Bauer, B. (2014). Analysis and Validation of AUTOSAR Models.

Kumar, A., Raghavan, P., Ramanathan, J., and Ramnath, R. (2008). Enterprise Interaction Ontology for Change Impact Analysis of Complex Systems. In *IEEE Asia-Pacific Services Computing Conference, 2008. APSCC '08*, pages 303 –309.

Kurpjuweit, S. and Aier, S. (2009). Ein allgemeiner Ansatz zur Ableitung von Abhngigkeitsanalysen auf Unternehmensarchitekturmodellen. *Wirtschaftinformatik Proceedings 2009*.

Lankhorst, M. (2012). *Enterprise Architecture at Work*. Springer-Verlag Berlin and Heidelberg GmbH & Co. KG, Berlin.

Lehnert, S. (2011). A review of software change impact analysis. *Ilmenau University of Technology, Tech. Rep*.

Matthes, F., Monahov, I., Schneider, A., and Schulz, C. (2012). EAM KPI Catalog v 1.0. Technical report, Technical University Munich.

MID GmbH (2014). MID Innovator for Enterprise Architects. in: http://www.mid.de/produkte/innovator-enterprise-modeling.html, accessed 15/04/2014.

Närman, P., Buschle, M., and Ekstedt, M. (2012). An enterprise architecture framework for multi-attribute information systems analysis. *Software & Systems Modeling*, pages 1–32.

Niemann, K. D. (2006). *From enterprise architecture to IT governance*. Springer.

Saad, C. and Bauer, B. (2011). The Model Analysis Framework - An IDE for Static Model Analysis. In *Proceedings of the Industry Track of Software Language Engineering (ITSLE) in the context of the 4th International Conference on Software Language Engineering (SLE'11)*.

Saad, C. and Bauer, B. (2013). Data-flow based Model Analysis and its Applications. In *Proceedings of the 16th International Conference on Model Driven Engineering Languages and Systems (MoDELS'13)*.

Saat, J. (2010). Zeitbezogene Abhängigkeitsanalysen der Unternehmensarchitektur. *Multikonferenz Wirtschaftsinformatik 2010*, page 29.

Tang, A., Nicholson, A., Jin, Y., and Han, J. (2007). Using bayesian belief networks for change impact analysis in architecture design. *Journal of Systems and Software*, 80(1):127148.

The Open Group (2012). *ArchiMate 2.0 specification: Open Group Standard*. Van Haren Publishing.

U.S. Department of Defense (2010). The DoDAF Architecture Framework Version 2.02. in: http://dodcio.defense.gov/dodaf20.aspx, accessed 15/03/2015.

von Knethen, A. and Grund, M. (2003). QuaTrace: a tool environment for (semi-) automatic impact analysis based on traces. In *International Conference on Software Maintenance, 2003. ICSM 2003. Proceedings*, pages 246–255.

The Intertwinement of Architectural Governance and Enterprise IT-Architecture

Enterprise IT-Architecture Viewed as Boundary Object from a Complex Adaptive Systems View

Marijn Janssen
Faculty of Technology, Policy & Management, Delft University of Technology, Jaffalaan 5, Delft, The Netherlands
m.f.w.h.a.janssen@tudelft.nl

Keywords: Enterprise Architecture, IT-Architecture, Governance, Architectural Governance, Organizational Networks, Public-private Networks, Business Networks, Boundary Objects, Complex Adaptive Systems.

Abstract: Organizations collaborate more and more in networks in which the boundaries between organizations are flux to changes. The collaborating organizations have various degrees of technology-readiness, capabilities, heterogeneous systems and processes which are developed over time. Despite these differences these organizations have to work together in a loosely coupled structure, where the overall process performance depends on the weakest chain. This paper explores the intertwinement of governance and enterprise IT-architecture (EA) by adopting a complex adaptive systems perspective and viewing EA as boundary object. Architects can act as boundary spanners and architectural models and documents as boundary artefact. Both help in creating shared understanding. Governance and EA are intertwined and need to be developed in concert to realize its value. Our analysis shows that architecture and governance are mutual dependent and influence each other.

1 INTRODUCTION

Organizations cooperate more and more in networks and are exchanging information with each other. Often this is donated as organizational networks (Provan and Milward, 2001), business networks (Short and Venkatraman, 1992), public-private networks (Grimsey and Lewis, 2007). The essence is that a number of organizations work together in a loosely coupled structure which is subject to variations. Organizational network arrangements are aimed at coordinating horizontal and vertical dependencies between agencies.

Often the collaborating organizations are heterogeneous in many aspects which complicates cooperation. Each of the organizations have their own systems and their IT-architecture which has been developed independently of the other organizations. Whereas collaboration requires interoperability at all levels including the technical, syntax, semantic and pragmatic level. the pragmatic layer refers to the context-sensitive aspect of meaning (Singh, 2002).

Enterprise IT-architecture (EA) is based on statements of how an enterprise wants to use its Information and Communication Technology (ICT) to accomplish its strategy and relate this to the vision of what ICT has to offer for the enterprise and how this should be realized. EA is an instrument to ensure that in business networks can be collaborated in an interoperability, a flexible and agile manner. EA can be used to align the own IT with the IT of other organizations and guide design decisions and new initiatives. As such, it is closely intertwined with *architectural governance* (AG). A lack of effective IT governance can easily result in failed development projects or lack of collaboration (Peterson, 2004). AG can be viewed as the processes and authority for decision-making concerning the use and development of EA.

Despite the significance of having governance the architecture function is often disconnected from the other parts of the enterprise. An EA identifies the main components of the enterprise, its information systems, the ways in which these components work together in order to achieve defined objectives and the way in which the systems support business processes. Architectures are meaningless if they are not adopted and used. Furthermore, architectures need to be flexible and agile to incorporate new technology. Enterprise Architecture and Governance (EAG) is an upcoming field which influence the use, success and adoption of new technology. The relationship between EA and AG has been given limited attention and are influenced and shaped by each other. In this paper this relationship is further explored. In the next section we explore the literature and conclude that

2 BACKGROUND

2.1 Organizational Networks

Networks have become a form of organizing in which multiple organizations collaborate together. These organizations are often autonomous and make their own decisions. Organizations cooperation in a business network typically differ in many aspects, including.

- Objectives and interests
- Organization structures,
- Degree of technology-readiness
- Heterogeneous type systems
- Set of standards and architectural choices
- Service level agreements
- Set of capabilities

These characteristics makes it often hard to collaborate in networks. In addition there might be differences like culture and challenges like unclear responsibilities which further complicate collaboration. Yet linking among organizations is vital, because the connect organizations to one another.

2.2 Enterprise IT-Architecture

Enterprise IT-architecture (EA) aims to bridge the gap between organizational and technology aspects by looking at various levels of abstraction and adopting different stakeholder views (Zachman, 1987). EA can describe or prescribe the dependencies among a large number of heterogeneous elements to ensure that these elements are addressed. In this way, EA can guide design decisions and provide direction for progressing towards a next generation infrastructure (Janssen et al., 2009).

In essence, EA should address the dependencies among a large number of heterogeneous elements at both the organizational and technical levels (Doucet et al., 2008). Traditionally, the purpose is to effectively align the strategies of enterprises with their business processes and the coordination of their resources (Zachman, 1987). EAs define and interrelate data, hardware, software, and communication resources, as well as the supporting organization required to maintain the overall physical structure required by the architecture (Richardson et al., 1990). The primary strength of the EA approach is that it has well-defined concepts and instruments to control and develop complex, technological systems. Nevertheless, much of the past research on EA can be criticized for taking a technologist view or being too abstract. Architecture models, principles and standards make up the content of EA (e.g. Simon et al., 2013, Janssen and Hjort-Madsen, 2007).

2.3 Architectural Governance

Architecture and governance are viewed as interrelated for a long time. Spewak (1992) introduces architecture as a process. Architecture as a process giving directions to the development of a system. In contrast, architecture as an instrument is a blueprint describing the relationships among the components in the system. The Architecture as a process view looks at how the architecture is used as an organizational routine to shape information projects.

Governance represents the framework for decision rights and accountabilities to encourage desirable behaviour in the use of resources (Weill, 2004). Architectural Governance is necessary for creating, designing, assembling, maintaining and exploiting EA. Once the decision to introduce a EA is made a coordinated set of governance mechanisms is necessary. Architects and managers will follow these governance mechanisms on a daily basis to guide EA development and maintenance.

In general there are three kinds of governance mechanisms: (1) decision-making structures, (2) alignment processes and (3) formal communications (Weill and Ross, 2005). *Decision-making structures* refer to the organisational committees and roles for decision-making. *Alignment processes* are management techniques for securing widespread and effective involvement in the architecture use, the making of architectural decisions and EA implementation. This includes the way the EA function is funded. *Formal communications* is about the communication between EA function and the other parts of the organization and should ensure a good relationships between the both of them.

A lack of understanding about how decisions are made, what processes are being implemented and what the desired outcomes are might disrupt organizational processes (Weill, 2004). Hence clear procedures and processes and other governance mechanisms are views as crucial. More communication generally means more effective governance (Weill and Ross, 2005).

2.4 Boundary Spanning

Organizational networks are underpinned by reciprocity, mutual trust and willingness to share information. Links among organizations and departments are vital to competitive advantages. These types of links among organization should occur at all levels. Boundary spanning can be done by persons who are connected to two organizational entities or by artefacts that are used by different entities as shown in figure 1.

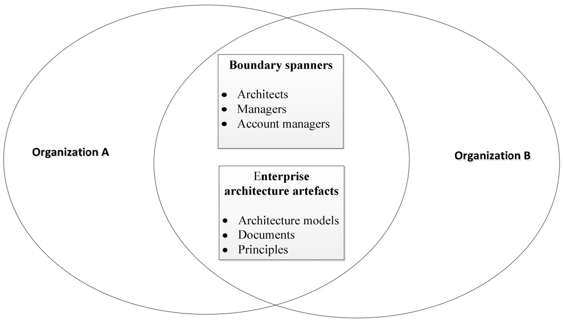


Figure 1: Boundary objects in enterprise architecture.

Boundary spanners are organizational members who link their organization with other organizations (Thompson, 1967). Architectures are typically boundary spanners as the link business and IT people. Boundaries spanners intermediate interactions with other department or organizations and are gatekeepers at the same time. As gatekeepers they select and filter information before distributing these in their own organizations. Boundary spanners should have knowledge for whom information is relevant and make decisions concerning the distribution of gathered information. Boundary spanners convey influence between the various groups and at the same time represents the perceptions, expectations, and values of the own organizations to this groups (Friedman and Podolny, 1992).

Boundary objects may include physical product prototypes, design drawings, shared IT applications, standard business forms, or even shared abstract constructs such as product yield (Levina, 2005). Boundary objects are referred to as conceptual or physical artefacts that reside in the interfaces among organizations (Gal et al., 2008). Boundary objects are plastic and provide interpretive flexibility, but with enough immutable content to maintain integrity and create shared understanding.

From a functional perspective, a boundary object is an artefact shared by a community of subjects like architectural documents, blueprints, but also systems used by different persons. Thee artefact are used by individual to reach their own goals. Such an object can be a physical object, but can also be intangible like a shared visions on how EA should be like. A boundary object must be "both plastic enough to morph to local needs and constraints of the several parties employing them, yet robust enough to maintain a common identity across sites" (Star and Griesemer, 1989 p. 393). Boundary objects help interacting organizations facilitate cross-organizational communication and form an organizational identity (2008), while they can also act as gatekeepers that selectively filter information between the organizations. Hussenot and Missionier (2010) depict them as bearers of compromises that promote cooperation between the stakeholders (p.274). Therefore, boundary objects should be able to bear various meanings assigned by different organizations while serving as a common reference point to the members of multiple organizations when they engage in mutual practice (Star and Griesemer, 1989).

For a boundary object to emerge, a new joint field of practice must be introduced (Levina and Vaas, 2005). For example, an architectural document or blueprint can be a joint field of practice. However, not every artefacts might become a boundary object in practice. Human agents in some organizations may not see its local usefulness or the artefact may fail to establish a common identity across all organizations (Levina and Vaas, 2005). Often effective governance mechanisms are necessary to ensure that persons and artefacts become boundary spanners. Little is known how boundary spanners and artefacts look like and about their effectiveness.

2.5 Complex Adaptive Systems

The basic idea of Complex Adaptive Systems (CAS) is that the level of analysis is at a lower level of aggregation. In other words the behaviour of a system is made up of interacting subsystems. Although there is no single uniform definition, an generally accepted definition of CAS is "*a system* *that emerges over time into a coherent form, and adapts and organizes itself without any singular entity deliberately managing or controlling it*" (Holland, 1996 p.10). Anderson (1999) found that complex adaptive systems can be summarized in terms of four properties; agents with schemata, self-organization, coevolving agents, and system evolution. Translated to organizational networks the four properties can be described as follows.

1. *Agents with Schemata*. An organization is made up of individuals, coalitions with have non-linear interactions. This dynamic behaviour needs to be understood. Architecture and governance influence the behaviour of individual agents to accomplish system level changes.
2. *Self-organization*. Agents are connected to and interacting with each other resulting in dynamic behaviour. By interacting with each other feedback loops are created with results in updates of the architecture and governance. In organizational networks there is no single coordinator directing these interactions, instead self-organization emerges from the interactions.
3. *Co-evolving Agents*. Individual agents are directed by others and follow their own interests. There is a continuous flux of changes and agents are continuously co-evolving with them. As a consequence of these feedback loops a small change might result in a whole chain of reaction.
4. *System Evolution*. New organizations and persons can enter and others can appear and the linkages among agents can change over time resulting in system evolution. New organizations or persons might become part of the network.

CAS can be used to predict that EA and AG influence each other and emerge over time. Although EA is designed the EA elements will be influenced by the governance. For example if governance does not require adherence to certain principles, then these principles will likely not be maintained and further developed. The recognition that the principles are necessary to accomplish certain benefits might result in the mandatory use of principles.

Organization theory often treat complexity as an independent variable, whereas in reality behaviour of complex systems is hard to predict because it is nonlinear (Anderson, 1999). Architecture and governance aspects might influence the stakeholders which in turn influence each other and the architecture and governance. Capturing the nonlinear outcomes of many interacting components is difficult (Casti, 1994).

3 RESEARCH METHOD

The goal of this research is to understand the relationship between EA and AG. The results will add to our knowledge concerning the design and management of architectures and accompanying governance mechanisms. Figure 2 shows the research model. The model is based on the mutual dependence between architecture, governance and organization performance. The constructs in the model can be extend based on the understanding within a case study. For example organizational is likely to differ per situation as organizations might have different objectives. Different objectives might in turn influence the EA and AG.

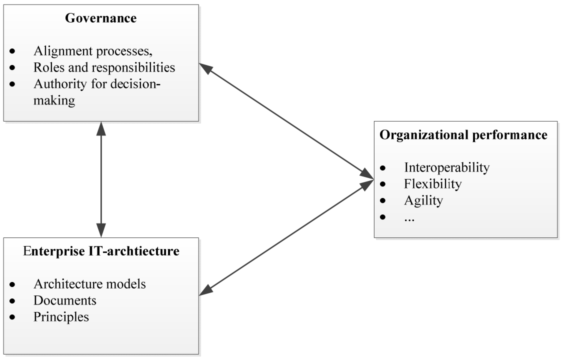

Governance
Alignment processes,
Roles and responsibilities
Authority for decision-making
Organizational performance
Interoperability
Flexibility
Agility
...
Enterprise IT-archtiecture
Architecture models
Documents
Principles

Figure 2: Research model.

We used a participative group session with representatives from various organizations to explore this complex relationship. DA group process can be a suitable research instrument as it allows participants to identify the critical dimensions and clarifying the meaning and subjective interpretation of each dimension using an interactive group process. Furthermore the discussion among participants can give more detailed insight. This allows to gain heterogeneous inputs which adds to the quality of the outcomes (Gustafson et al., 1973). Furthermore, the group session enabled the collect the data within the short time-frame of half a day. In this way the time of the participants was limited, while at the same time allowing the participants to share their opinions, arguments and evaluations of the various options and facilitating learning.

During the groups session 7 interorganizational networks were explored and ranked on the dimensions of governance and architecture. The session was facilitated the author and organized as part of a lecture series. The session was held in November 2013. During the session the main steps followed were:

1. *Introduction*
2. *Background:* background about architecture and governance was presented.
3. *Assessment*: All participants were asked to outline historical development and to make a qualitative assessment of their architecture and governance over time to gain indepth understanding of the interplay between governance and architecture.
4. *Scoring and positioning*. Each of the participants positioned the results on a whiteboard displaying the matrix as shown in figure 2.
5. *Discussing the results*. Participants were asked to discuss the results.
6. *Closing*

The use of the architecture and governance dimensions resulted in the creation of four quadrants and each quadrant was given a name. The naming and explanation of the quadrants should help the session participants to ensure the right positioning. The matrix as shown in figure 3 was used to map the developments and results.

1. *Ad hoc*: both governance and architecture are ill-developed. There is hardly any structured information exchanges among the organizations in the network.
2. *Technocratic*: In this quadrant architecture is well-developed and leading, but governance is not mature. The focus is often on developing the EA and less on the actual use.
3. *Governed*: In this quadrant the governance is mature, whereas architecture is immature. There is much interaction among organization, but there is limited agreement about architectural standards, principles and models.
4. *Architecture Governance*: In this quadrant both architecture and governance is well-developed and can strengthen each other. Often this is viewed as the normative ideal.

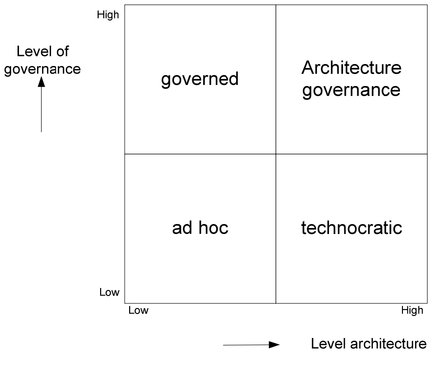


Figure 3: Intertwinement of governance and architecture.

4 CASE STUDY

To understand the dependence between architecture and governance a complex interorganizational network was studies that evolved over time. The SUWI network (Structuur Uitvoerings-organisatie Werk en Inkomen) is aimed at make sure that there are no persons without a job or income. For this purpose information needs to be shared and information collected by one party should be shared with the other organizations. In the SUWI network many organization are participating, including;

- Centrale organisatie Werk en Inkomen (CWI, www.CWInet.nl);
- Uitvoeringsinstituut Werknemersverzekeringen (UWV, www.uwv.nl);
- Sociale Verzekeringsbank (SVB, www.svb.nl);
- Inlichtingenbureau (IB, www.inlichtingenbureau.nl)
- Bureau Keteninformatisering Werk en Inkomen (BKWI, www.bkwi.nl).
- Municipal social security departments
- Private Service providers

There are over 400 municipalities who's social departments can be involved. In addition other organizations can be involved responsible for providing services to citizens. An example are the reintegration organization that support unemployed to become employed again.

The goal of this network is twofold, 1) to make sure that there are no persons without a job and ensure that unemployed persons will employed soon and 2) to make sure that every persons has a minimum amount of income. In addition all the partners have their own goals which are often determined by law.

When information about persons is shared with other organizations in the network this organization should only obtain the information relevant for them and privacy regulations should be met. In this transfer process information can be lost. The transfer requires information sharing technology and clear agreements about the meaning of the information and who is responsible for the information.

The development of the network initially focused on creating the platform for information sharing. Nevertheless although the focus was on the architectural elements, governance played a crucial role. There were many interactions among the members. For example the harmonization of terms to ensure that the same meaning is given requires extensive governance mechanisms. Although the focus was on making the technology first, this was always guided by governance mechanisms. As such the participants in the session had a hard time positioning the efforts in figure 2. Whereas some participants were initially convinced that architectures was started they started doubting this after the discussions. At the end there was consensus that both efforts influenced each other. Although the driver was technology, both architecture and governance were necessary to make this work in this interorganizational network.

When creating the network many architectural pictures and documents were used as boundary objects. Nevertheless some of them proved to be not suitable as boundary objects as 1) they were hard to understand or interpret by others, 2) provide too much detail or 3) used terms that did not belong to the other community. As such dedicated attention should be given to architecture objects that should function as boundary objects between the organizations. They should be plastic enough to provide interpretive flexibility and adapt to the needs of each of the organizations involved. This does not mean that they need to be abstract. They can also be detailed like the realization of secure data exchange. This needs to be detailed to ensure that each of the organization had a shared understanding. Nevertheless principles need to be abstract to be reframed in different context and allow to direct the organizations. The creation of architectural artefact as boundary objects was found to be crucial for making architecture work. Furthermore the creation of methods standardization is crucial for making the governance work.

CAS recognizes that a system can be guided using relatively simple principles (Holland, 1996). The typical example of CAS is a flock of birds in which individual birds follow the principles of having a certain distance from each other (the V-shape) and switch position regularly. In the case study the recognition that architectural principles are necessary to accomplish certain benefits might result in the mandatory use of principles. One person commented "*the development of several systems without having control and maintenance interface from the start resulted in the acknowledgement by management that control and maintenance interface should be part of the user acceptance test*". This resulted in the development of a mandatory principles outlining this.

Organization performance should be viewed as relatively broad, in which aspects like project failure, lack of flexibility, agility, interoperability etc. play a major role. Governance and architecture can influence these aspects. How governance and architecture is operated and used is determined by its organizational members which determines in this way system level performance.

5 CONCLUSIONS

There is a high interdependency between governance and enterprise IT-architecture which makes it complex to analyse. Governance and EA can strengthen each other, but failure of having governance in place can result in ineffective EA. EA and AG are mutual dependent which can be explained form a Complex Adaptive Systems (CAS) lens in which feedback loops play a major role.

EA and governance are often said to be interrelated, but more comprehensive conceptualizations of how the process unfolds and contributes to the organization performance is ill-understood. The complexity originating from the many persons involved and large and heterogeneous system landscape, the multilevel nature, non-linear and emergent behaviour makes it difficult to conceptualize and understand the relationship between architecture and governance. We used a CAS lens to conceptualize these kind of processes better and how these results in success or failure. By adoption a CAS lens emergent phenomena, interdependencies and interactions can be conceptualized. In this way the understanding of how individual agents influence system level becomes clearer and interventions can be developed to deal with it as was done in this research.

In addition we analysed the boundary spanners and objects. Architects can act as boundary spanners and architectural models and documents as boundary artefact. It was found that architectural pictures and documents can be used as boundary objects if they are plastic enough to adapt to the different context and provide interpretive flexibility. Boundary objects can be very detailed (i.e. blueprints and cookbooks describing how something needs to be done) or abstract (i.e. principles given direction in different context).

Furthermore, CAS accounts for that architecture principles and elements and governance mechanisms might working in a certain situation might not be suitable for another situation. More research is necessary in adopting the CAS lens and viewing architecture as boundary objects.

REFERENCES

Anderson, P. 1999. Complexity theory and organization science. *Organization Science,* 10, 216-232.

Casti, J. 1994. *Complexification: Explaining a Paradoxical World Through the Science of Surprise,* New York, HarperCollins.

Doucet, G., Gøtze, J., Saha, P. & Bernard, S. 2008. Coherency Management: Using Enterprise Architecture for Alignment, Agility, and Assurance. *Journal of Enterprise Architecture,* 4, 9-20.

Friedman, R. A. & Podolny, J. 1992. Differentiation of boundary spanning roles: labor negotiations and implications for role conflict *Administrative Science Quarterly,* 37, 28-47.

Gal, U., Lyytinen, K. & Yoo, Y. 2008. The dynamics of IT boundary objects, information infrastructures, and organisational identities: the introduction of 3D modelling technologies into the architecture, engineering, and construction industry. *European Journal of Information Systems,* 17, 290-304.

Grimsey, D. & Lewis, M. K. 2007. *Public Private Partnerships: The Worldwide Revolution in Infrastructure Provision and Project Finance,* Cheltenham, UK, Edward Elgar Publishing Limited.

Gustafson, D. H., R.K., S., L., D. A. & G.W., W. 1973. A comparative study of differences in subjective likelihood estimates made by individuals, interacting groups, Delphi groups, and nominal groups. *Organizational Behavior and Human Performance,* 9, 280-291.

Holland, J. H. 1996. *Hidden order. How adaptation creates complexity,* Reading, MA, Addison Wesley.

Hussenot, A. & Missonier, S. 2010. A deeper understanding of evolution of the role of the object in organizational process: The concept of "mediation object". *Journal of Organizational Change Management,* 23, 269-286.

Janssen, M., Chun, S. A. & Gil-Garcia, J. R. 2009. Building the next generation of digital government infrastructures *Government Information Quarterly,* 26, 233-237.

Janssen, M. & Hjort-Madsen, K. 2007. Analyzing Enterprise Architecture in National Governments: The Cases of Denmark and the Netherlands. *In:* SPRAGUE, R. (ed.) *40th Annual Hawaii International Conference on System Sciences (HICSS'07).* Big Island, Hawaii: IEEE.

Levina, N. 2005. Collaborating on Multiparty Information Systems Development Projects: A Collective Reflection-in-Action View. *Information Systems Research,* 16, 109–130.

Levina, N. & Vaas, E. 2005. The Emergence of Boundary Spanning Competence in Practice: Implications for Implementation and Use of Information Systems. *MIS Quarterly,* 29, 335-363.

Peterson, R. 2004. Crafting Information Technology Governance. *Information Systems Management,* 21, 7-22.

Provan, K. G. & Milward, H. B. 2001. Do Networks Really Work? A Framework for Evaluating Public-Sector Organizational Networks. *Public Administration Review,* 61, 414-423.

Richardson, L., Jackson, B. M. & Dickson, G. 1990. A Principle-Based Enterprise Architecture: Lessons From Texaco and Star Enterprise. *MIS Quarterly,* 14, 385-403.

Short, J. & Venkatraman, N. 1992. Beyond business process redesign: redefining Baxter's business network. *Sloan Management Review,* 34, 7-21.

Simon, D., Fischbach, K. & Schoder, D. 2013. An Exploration of Enterprise Architecture Research. *Communications of The Association for Information Systems,* 32, 1-72,.

Singh M. P. 2002. The pragmatic web. *IEEE Internet Computing,* 6, 4-5.

Spewak, S. H. 1992. *Enterprise architecture planning. Developing a blueprint for data, applications and Technology,* New York, John Wiley.

Star, S. L. & Griesemer, J. R. 1989. Institutional ecology, 'translations' and boundary objects: amateurs and professionals in Berkeley's museum of vertebrate zoology, 1907-1939. *Social Studies of Science,* 19, 387-420.

Thompson, J. D. 1967. *Organizations in Action,* New York, McGraw-Hill.

Weill, P. 2004. Don't Just Lead, Govern: How best Performing Organisations Govern IT. *MIS Quarterly Executive,* 3, 1-17.

Weill, P. & Ross, J. W. 2005. A matrixed approach to designing IT governance. *MIT Sloan Management Review,* 46, 26-34.

Zachman, J. A. 1987. A Framework for Information Systems Architecture. *IBM Systems Journal,* 26, 276-292.

A Relation-Algebra Language to Specify Declarative Business Rules

Lex Wedemeijer
Department of Computer Science, Open University, The Netherlands, Valkenburgerweg 177, Heerlen, The Netherlands
Lex.Wedemeijer@ou.nl

Keywords: Declarative Business Rules, Relation Algebra, Modeling Language, Metamodeling, Rule Compliance.

Abstract: Business rules that apply within a business context must be formulated in a comprehensible way to allow validation by their stakeholders, but at the same time they must be specified with enough precision to assure their correct implementation in computer applications. These opposing demands of business rule modeling are not easily reconciled. Formal rule modeling languages may be exact but they are often lacking in understandability, whereas controlled natural languages are more easily understood but generally fall short in exactness. We use Relation Algebra as the foundation to set up a controlled language for declarative business rules that is compatible with practical demands, such as laid out in the Business Rules Manifesto. Our version of controlled language comprises just five language statements that are orthogonal by design, which makes for a language that is suited for use by novice business rule modelers. The language lets users set up a business vocabulary that stakeholders can understand, and it allows to specify business rules about the objects in the vocabulary in a comprehensible if-then syntax. Rules expressed in our language are precise enough to permit the automatic generation of a prototype information system which is guaranteed to comply with the rules. Stakeholders can explore this prototype to verify the vocabulary, and to check whether the specified rules are valid and match their original intent of the business context. We show how we can ascertain correctness of our language and metamodel, by adopting a reflective approach and subject our context to rule analysis and specification, just like any other business context. It provides us with a prototype system that lets us explore the rules about rules, and validate the rule compliance.

1 INTRODUCTION

Business rules play an important role in day-to-day business operations and supportive IT applications. Declarative rules restrict what states are permitted in the business, and which operations may be executed by employees and information systems of that business (Hay, Healy 2000).

There is consensus that the business rules should be validated by stakeholders in the organization to ensure their overall correctness and coherence (Business Rules Manifesto 2003). Therefore, rules must be expressed in a way that a target business audience clearly understands. But to use those very rules in software applications calls for exact specifications and computer precision. This poses contradictory demands: comprehensibility for lay users, but perfect exactness for programmers and applications.

The prime deliverable of rule-based design is a compliant database application. In practice, the informal rules of business behaviour are rephrased and transformed in a chain of handovers until their encapsulation in an enterprise information system (figure 1). At each point in the chain, requirements are translated into yet another language, a process which is prone to misinterpretations, loss of detail, and other problems, even in the presence of a validated vocabulary (Bajwa et al., 2011).

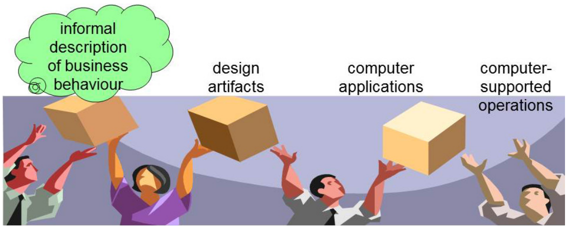

informal description of business behaviour
design artifacts
computer applications
computer-supported operations

Figure 1: Chain of handovers of business rules.

To reduce the need for translation, we propose a simple language founded on proven theory to cover the major part, if not the entire chain of handovers.

The business rules considered in this paper are declarative: there is no procedural dependence or hidden sequencing. The rules are also invariant: they concern persistent states only, not some transient situations that exist for just a brief moment in time, e.g. only while a data transaction lasts. This differs from IT-approaches like the Event-Condition-Action (Poulovassilis et al., 2003) paradigm, or Communicating Sequential Processes (Hoare, 1985; Wedemeijer, 2012). From a business point of view, the ECA type of rules have a technical ring, and their relevance is experienced as vague, difficult to retrace, and hard to explain (Andreescu, Mircea 2014).

The paper outline is as follows. Section 2 discusses some contemporary languages for declarative business rules, and design considerations for our language. Section 3 describes the proposed language. The syntax of each basic statement is depicted as a railroad diagram, and we explain core ideas. Section 4 puts the language to work, by describing features of supportive design- or prototype environments. Such an environment can be regarded as a business context having its invariant rules captured. In section 5 we pursue this idea by developing a meta-model of the language. Section 6 presents conclusions.

2 RELATED WORK

Business rule languages must be comprehensible for business workers on the one hand, and faultlessly exact for computer applications on the other (Bjekovic, Proper 2013). Numerous languages to express declarative business rules exist (Kardasis, Loucopoulos 2004). Our discussion of languages is restricted due to lack of space.

2.1 Declarative Rule Languages

On one side of the spectrum of languages to express business rules are natural and semi-controlled languages. Prominent Semantics of Business Vocabularies and Rules (Object Management Group 2008). One of its derivatives is RuleSpeak, 'a set of guidelines for expressing business rules in concise, business-friendly fashion using structured natural language' (Ross, Lam 2011). Another derivative is Attempto Controlled English (Fuchs et al., 2008).

These approaches rely on business vocabularies, also called 'fact models', in order to capture the true meanings and definitions of business data. Hence, comprehensibility and business focus is a strong point. However, controlled languages still permit a large variety in phrasing, and lack uniformity. As a result, rules are not always concisely and clearly expressed, making validation difficult and leaving room for interpretation, two fatal shortcomings for IT implementation (Weigand et al., 2011).

Other standards based on SBVR are FBM (FBM Working Group 2011) and Object-Role Modelling (Halpin, 2011). Both standards depict conceptual models in the customary way, and then depict the constraints visually. As a result, the diagrams with constraints become quite confusing, and they are barely intelligible for lay users.

A middle field is languages that aim to describe enterprise architectures, stakeholder concerns, goals and business rules (Quartel et al., 2009). Generally, these languages are not geared to capture rules, and are too high-level to allow validation by business stakeholders, or implementation in IT-systems.

On the other side of the spectrum are languages with an IT-provenance, such as UML- and XML-based languages or DTD's. Many of these languages are 'rich', meaning that a business feature may be captured in a variety of ways (Lamrani et al., 2013). Hence, it requires a thorough knowledge of implementation details to disclose the business relevance of an implemented rule (Beckner, 2014). Andreescu and Mircea (2014) remark on the reluctance to use OCL in the early design phases, when IT specialists need to cooperate with business people.

RuleML is an evolving family of XML-based languages (Boley et al., 2004). Semantic Web Rule Language, SWRL for short, achieves an expressive power superior to our language in some areas, e.g. to specify derivations, numeric and time calculations (Horrocks et al., 2004). SWRL also includes the Horn-clause syntax for rules, a strong point that which we will employ in our language. Nonetheless, the IT-orientation and notational complexity of SWRL, and XML-based languages in general, make them unsuitable for an average business user or novice designer (Akbari et al., 2103).

We conclude that (controlled) natural languages may capture business rules in a comprehensible and validatable manner, but not precise enough for computer applications. Formal rule modeling languages or general IT languages may be exact enough, but they lack in understandability.

2.2 Language Considerations

With the above in mind, our language for business rules was devised with the intention to:

- ensure comprehensibility for business people by relying on business vocabulary (terms and phrases of the business context).
- ensure that business workers can understand and validate their rules, and so minimize the need for back-and-forth translation of rules.
- ensure orthogonality of the language, so that

features are always expressed in just one way, and so avoid the problems of 'rich' languages.

- ensure exactness of rules, by founding them on rigorous mathematical theory; we opt for binary Relation Algebra (Maddux, 2006).

To prevent trivial but cumbersome errors in data entry, we prefer names and identifiers to be case-insensitive. Also, leading and trailing spaces should be avoided as much as possible.

2.3 Way of Working

Rule design may be conducted in a progressive way of working (figure 2).

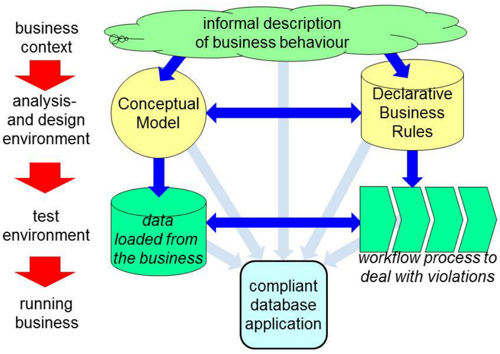

business context
analysis- and design environment
test environment
running business
informal description of business behaviour
Conceptual Model
Declarative Business Rules
data loaded from the business
workflow process to deal with violations
compliant database application

Figure 2: Way of working in rule-based systems design.

The approach starts at business behaviour, which in most cases is only informally understood.

In the analysis and design phase, a business model of concepts and relations is created capturing the relevant parts of the business vocabulary. And, very important, the declarative rules are captured.

In the test environment, data for concepts and relations is loaded incrementally to test whether the predicted rule violations emerge. Rule enforcements are specified, determining how workflow processes should deal with rule violations in practice.

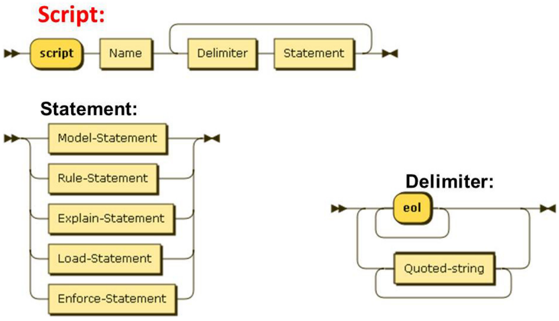


Figure 3: Railroad diagram for script and statements.

3 PROPOSED LANGUAGE

Our language provides five statement types that a designer may use in the specification of a business context. A railroad diagram of the overall language set-up is shown in figure 3.

Statements in a script may appear in arbitrary order, to suit an incremental, step-by-step, top-down, big-bang, modular, or any other preferred approach of the designer. The statements are uniform in make-up: first a reserved language imperative, identifying name(s) next, and then the further particulars.

3.1 Model

The model statement defines the concepts and binary relations that are part of the business vocabulary (figure 4). It sets up the structure of concepts and relations that the designer considers to be important.

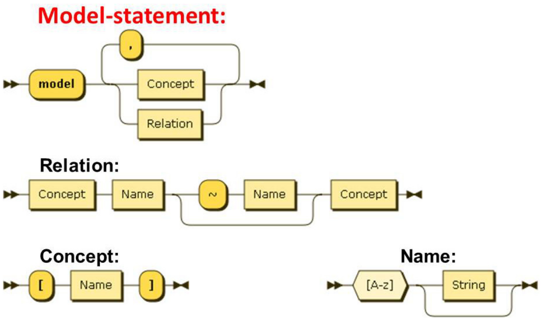


Figure 4: Railroad diagram for the model statement.

Concepts have unique names, enclosed in square brackets for clarity, and starting with a letter.

Relations are uniquely identified by a colloquial name to call the relation by, plus the names of its domain and range concepts. In addition to the colloquial name for the relation itself, another name may be provided for the inverse relation, indicated by the ~ inversion symbol. Uniqueness requirements for the relation name also apply to the inverse name.

We prefer the infix style of notation for relations. It enhances readability and prompts designers to pick self-explanatory relation names. Technically speaking, prefix or other styles are equivalent.

A script may contain multiple model statements, so that concepts and relations can be incrementally introduced. And because concepts are easily deduced from relation domains and ranges, a designer may even forego the explicit modelling of concepts.

By definition, it is impossible to violate a model. All true facts observed in the business context, either fit perfectly in the structure, or they are irrelevant. If some business fact is relevant but still cannot be expressed as atoms or tuples, then the structure is wrong: it is an inadequate model of the business context. Thus, a model may be regarded as a set of structural rules. As no other rules apply to it, we call this an Unconstrained Conceptual Model.

3.2 Rule

We are now in a position to specify 'behavioural' rules that business stakeholders ought to live by. These rules should always evaluate to being satisfied, but in a running business, they may temporarily be violated. The implication is not, that the model is wrong. Rather, the business stakeholders should take action to remedy the violations.

The combination of model and rule statements is sufficient for business rule analysis and design. The joint deliverable may be called a Conceptual Model, and a good designer will make sure that it meets the usual quality requirements, such as completeness, and consistency of its rules (Moody, 2005).

To emphasize the behavioural aspect of rules, a core position is given to the rule keyword 'must' in the rule statement. The idea, in accordance with the ideas of RuleSpeak (2014), is to help users grasp the rule intent: guiding the business behaviour and have people refrain from violating the rule.

Each rule comes with a unique rule identifier, starting with a digit 0.9. Other statements can refer to the rule by way of this identifier, and it also comes in handy when violations are to be reported.

3.2.1 Simple Rules: Cardinality Constraints

A single relation may already be subject to a simple rule, i.e. cardinality constraints may apply. For and understandability, our language provides keywords to express cardinalities and common combinations.

For instance, the keyword 'function' means that a relation must be univalent and total. In addition, we provided keywords for ruletypes of homogeneous relations. And although simple ruletypes usually apply to simple relations, a compound expression may also be subjected to this kind of rule.

Combining disparate cardinalities under one heading defies the idea of having a unique identifier for each distinct rule. The designer should decide whether or not to combine rules, depending on how the user community understands these rules and deals with possible violations.

Notice that simple ruletypes are syntactic sugar: all simple constraints are perfectly expressible in mathematically equivalent compound rules. We include the rule keywords in our language for the sake of simplicity. In practice, it makes little difference: a rule is referenced only through its identifier, independent of the mathematical formulation.

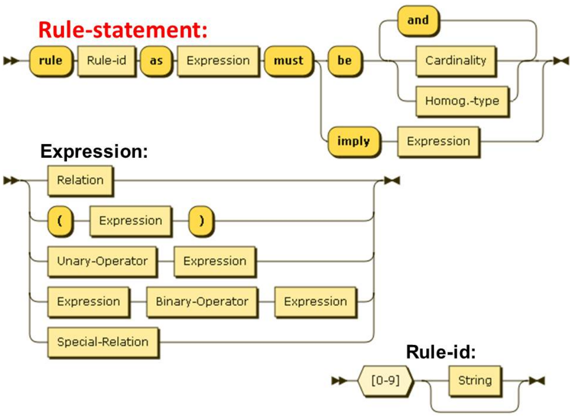


Figure 5: Railroad diagram for the rule statement.

3.2.2 Compound Rules: If-then Phrases

The real benefit of Relation Algebra is found in its ability to formulate compound rules in the concise yet straightforward way of normalized Horn clause format (Horrocks et al., 2004):

$$\text{antecedent} \Rightarrow \text{consequent}$$

Both the antecedent and consequent are binary relations, either a plain relation of the Unconstrained Conceptual Model or a compound expression, and must have the same concepts for domain and range. This format is easily translated to a semi-formal if-then sentence (1), into which we like to include the important rule keyword 'must':

IF the antecedent is confirmed,
THEN MUST the consequent be confirmed (1)

We define a rule violation as: a pair in the antecedent, but absent from the consequent expression. With this definition, the text becomes:

IF a pair is present in the antecedent,
THEN MUST the pair be in the consequent

The Horn-clause format easily pronounces as 'if... then must...', but the mathematical expressions in a rule may be quite complex, as seen in the railroad diagram of figure 5. Both expressions in the Horn clause may be a relation of the Unconstrained Conceptual Model, or may combine several relations using unary and binary operations. Special relations and constants may also be included, as explained below. It requires skills and business knowledge to translate the complex expressions into terms that the user community can understand. Better still is to avoid complex expressions altogether, and find easy-to-explain, natural rule assertions to begin with.

3.2.3 Special Relations and Constants

Complex expressions in rules may call for special relations. Our language provides a number of them, such as entire Cartesian Product, the empty relation, and the identity relation on a concept.

Expressions may also contain constant values or literals. Such values act as atoms or tuples in the rule expressions, but they need not be on record as they do not necessarily represent true business facts. Constant values in rules may force certain tuples to be on record. For instance, the rule 'the president of the USA must be a citizen', implicitly assumes that a nation named 'USA' is recorded. If we eliminate the constant by rephrasing the rule to 'the president of a nation must be a citizen of that nation', then the empty database no longer violates it. In general, an empty database never violates a rule if no constants are involved in the rule (Decker, Martinenghi 2006).

3.3 Explain

True business relevance means that each node, edge and clause in the specifications, can be clearly explained for, to, or even by the business workers. To help the audience grasp the detailed meaning and structure, explanatory texts are helpful (figure 6).

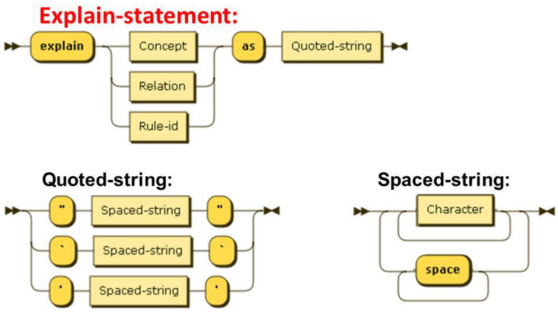


Figure 6: Railroad diagram for the explain statement.

An explain statement addresses either a concept, a relation, or a rule, each of which comes with its unique identifier. The explanatory texts do not alter the contents of the model or the rules and violations and therefore any number of explain statements may be given for a single concept, relation or rule. The aim is to help users in understanding both the details and the overall structure of the model.

3.4 Load

An important means to put a model and rules to the test is by loading data and check for rule violations. The ability to load data is also useful when a design is demonstrated to the business stakeholders.

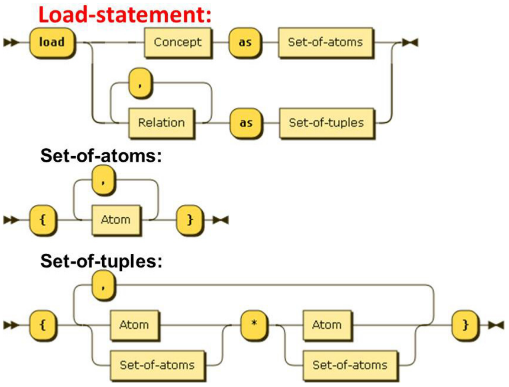


Figure 7: Railroad diagram for the load statement.

The load statement places sets of atoms, delimited by curly brackets { .. }, into a concept extension, or sets of tuples in relation extensions (figure 7).

Loading of data is not obligatory, but if data is loaded, then entity integrity and referential integrity is required (Date, 1981). At design time however, there is no need to worry about this, because it can be automatically ensured at load- and runtime.

3.4.1 Specifying Data

Like concepts, the atoms in our language are self-identifying: an atom is fully specified by its name, which is merely a text string, plus (the name of) the concept it belongs to. We do not distinguish between atom, atom-name, atom-value, or identity, distinctions that are hard to explain to lay users. Moreover, the atoms and their distinguishing names, or id's, or values must be linked tightly. Links that, on a meta-level, prove to be isomorphisms, or almost so. We think that such intricacies are better avoided.

Writing lots of data for loading is boring, and prone to typing errors. Our language provides two shortcuts. First, several relations may be loaded at once, provided of course that the domain and range are identical for all of them. Second, instead of specifying one tuple at a time, a set of tuples can be specified in one go, by combining a set of atoms from the domain with a set of atoms of the range.

3.5 Enforce

Enforcement is how runtime violations of the rules should be dealt with from the business point of view. In realistic business environments, enforcement may range from 'avoid violation at all cost' to 'comply or explain' or even 'ignore all violations'. From the IT perspective however, there are just three main strategies called 'projector', 'rejector', and 'producer' (Dietz, 2008).

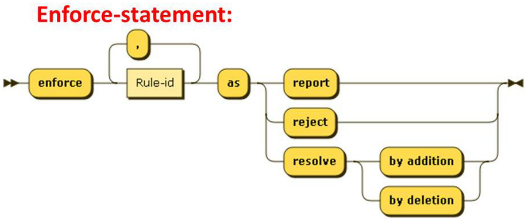


Figure 8: Railroad diagram for the enforce statement.

Our language provides for variants for all three variants. Figure 8 depicts the railroad diagram for the enforce statement in our language.

Enforce is not an incremental statement: a rule is subjected to one enforcement strategy at most, and specifying multiple enforcements for one rule has no use. If no enforcement option is specified for a rule, then the 'report' strategy applies by default.

3.5.1 Report

We call 'report' what the literature is referred to as 'projector'. This strategy for a rule means that after some edited data is committed, the rule is assessed, meaning that all violations are 'projected' into a special database table. Next, that listing is reported to the stakeholders. Notice that all violations ought to be reported, not just the new ones caused by the latest data edit.

Generally speaking, the 'report' strategy is easy to understand and robust to implement, which is why it is the safe choice for any business rule. For this reason too, this strategy is the default at load time.

3.5.2 Reject

The 'reject' type of enforcement strategy means that the rule must be checked prior to committing a change of data. If the change would result in a new violation for this rule, then the change should be rejected offhand, and the data not recorded.

The assumption underlying reject is that data violating this particular rule cannot even be valid in the business context. Rejection may be bothersome for business workers because this assumption is sometimes wrong, so that perfectly valid data is rejected for a bad reason.

There is another loophole: data may actually be in conflict with the rule, but if by coincidence the violating tuple is already present for another reason, then the erroneous data can be recorded nonetheless.

3.5.3 Resolve by

We introduce 'resolve' as final enforcement strategy for rules, referred to in the literature as 'producer'. The idea here is that sometimes in the business context, there is only one viable way to resolve a violation. And if the solution is known, why not let the computer apply it automatically?

We recall our definition of violation as a pair in the antecedent expression of a Horn clause, but not in the consequent. Hence, adding the offending pair into the consequent is a straightforward solution, and this is exactly what the enforcement strategy 'resolve by addition' intends. The strategy called 'resolve by deletion' takes the opposite tack and bluntly deletes the pair from the antecedent. As expressions in general cannot be edited, a necessary restriction is that the expression being edited must be a relation of the Unconstrained Conceptual Model.

A data-edit transaction is produced and in fact, this may result in the violation of some other rule. The automatic transition may even be rejected by another rule, or a subsequent transaction may be produced, and so on, potentially creating deadlocks or interminable loops. While compliant with the theory of Relation Algebra, this strategy goes beyond our context of invariant, i.e. state-oriented business rules. In defining our language, we did not investigate these effects nor have we tools for the rule designer to control them.

4 LANGUAGE ENVIRONMENT

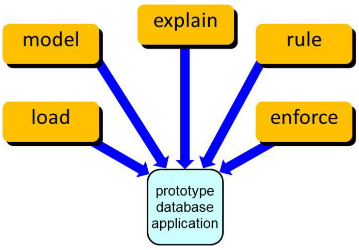


Figure 9: Contributing to the generated prototype.

The prime deliverable of rule design is a working database application that assures rule compliancy. Putting our rule language to work requires a design- and runtime environment in which all five of our statements contribute to that deliverable (figure 9). As we strictly adhered to the sound theory of Relation Algebra, the generated prototype is guaranteed to comply with the invariant rules in the script.

4.1 Design Time Interface

A design time interface should support a designer to create, expand, refine and correct her script, and also to save the script to continue work at a later time.

A graphic display of the Unconstrained Conceptual Model, with drag-and-drop and rearrange features to uncluttered the diagram, is a wonderful help in composing and understanding. Still, it is the model, the rule statements and explanations that should be at the core of the design effort, not the diagram.

The rule statement calls for a smart formula editor, with an option to link each rule expression to corresponding nodes or edges in the model diagram. Various flags would also be desirable, such as flags for faulty expressions, rule inconsistencies, potential simplifications of rules, or unconstrained relations.

The explain statement can well be supported by providing text editing functions in a mouse-over of the diagram.

The load statement needs generous support to let a designer include a full load of initial data in the script, and integrity should be taken care of automatically, at load time at the latest. Copy and paste of realistic data acquired from the business arena would be greatly appreciated. Automatic generation of datasets in compliance or violation of a specific rule, would also be desirable.

Lastly, no special support appears to be required for the enforce statement at design time.

4.2 Load Time Interface

At load time, all data in the script should be loaded into the Unconstrained Conceptual Model. Only the 'report' enforcement strategy is feasible at this time, to prevent undesired outcomes or even deadlocks caused by 'reject' or 'resolve' types of enforcement.

But data integrity must be made to hold. Referential integrity holds that each tuple refers to atoms on record in the domain and range, respectively. One option is to apply 'cascading delete', i.e. ignore all tuples that refer to a unrecorded atom. The opposite option is more attractive, i.e. to automatically insert the domain and range atoms of all tuples. Entity integrity holds that duplicates of an atom or tuple already on record, should not be loaded.

Once data is loaded into the Unconstrained Conceptual Model, only then should the rules be checked and violations reported. The report should enable the designer to trace each violation: what rule is violated, which atoms and tuples play a part.

A smart designer selects her data for loading in such a way that each violation is clearly understood, explained, and repaired. If a violation cannot be understood, then either the loaded atoms and tuples make no sense in the business, or the rule itself is in doubt. Or if particular violations can only be repaired by rigorously deleting data, then apparently some rules are contradictory. In any case, business people should be consulted to clarify the issue.

4.3 Runtime Interface

The script captures the rules of the business context, and a computer interface serves to confront the business community with their rules. Conveniently, it makes little difference how a designer organizes her statements in the script, because the business users are not exposed to the script directly.

A minimal requirement is a browse-and-explain interface. This should help users understand their exact business rules and violations. It ought to depict uncluttered diagrams of the entire Conceptual Model or parts thereof. It should display relevant explanations for all the concepts, relations and rules in the diagram. Also for each rule a complete list of all persistent violations must be provided, with explanations and traces how each violating tuple is determined from the corresponding Horn-clause formula. From there, the interface should support drill-down features to scrutinize partial populations or even individual instances in the diagrams.

Second, a demonstration interface is appreciated to emulate a workflow case in a (partial) business process. A series of predefined datasets is loaded in sequence, showing the emergence and subsequent resolution of violations as new data is being entered.

Adjusting rule enforcements on the fly in the runtime interface is still better. This would allow experimentation how to deal with rule violations and to probe the effects on the workflow processing.

The real benefit of our language is to generate a rule-compliant runtime prototype application with full data edit capabilities. Business people can then put that prototype to the question by entering all conceivable kinds of business data, view the responses by the system, and come to understand the workflow processes for dealing with the violations of their business rules.

5 LANGUAGE METAMODEL

So far, we discussed the modelling of an arbitrary business context, its vocabulary and rules. In this section, we change the perspective and select 'rule design' for our business context.

5.1 Metamodel

What is the business vocabulary of design? What are its rules? If we could specify this special context in perfect detail, and if a compliant tool environment would be available, then we could use these... to generate the prototype system for rules design, in a truly reflective approach.

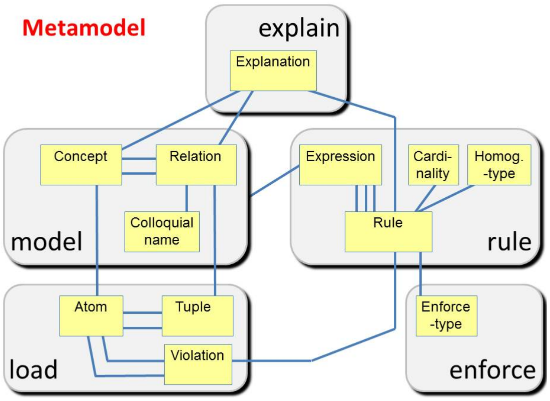


Figure 10: Metamodel of the language (conjectured).

The idea, although not new (Schön, 1992), is still worthwhile to pursue. Having outlined the relevant business context in section 3, we present a conjectured metamodel in figure 10. In the diagram, we use a freehand style, and we omitted the meta-relation names. Interestingly, the statements of our language can be associated with five distinct areas in the metamodel.

Evidently, the metamodel comes with its own set of rules constraints on the metamodel concepts, associations, and contents. Without being exhaustive we outline some major rules per area.

5.2 Meta-rules and Enforcement

Entity integrity is an intrinsic demand of relation algebra, and it applies in this metamodel as well. Duplicate atoms of a concept or duplicate tuples in a metamodel association are unacceptable at any time, This integrity demand can be enforced as 'reject', in other words: duplicate entries are simply ignored.

In the 'model' area, it is compulsory that each relation has exactly one domain concept, one range concept. Referential integrity proclaims that both concepts must be present in the extension of the Concept concept. This rule is easily enforced as 'resolve by addition', meaning that missing concept names are automatically inserted. Moreover, the domain concept, range concept, and colloquial name together must uniquely identify the relation. And if a relation's inverse name is given, then it must adhere to the same uniqueness demand.

In the 'rule' area, a simple rule is associated with one expression and one (or more) cardinality- or homogeneous-ruletype. The compound rules, i.e. those in Horn clause format, are associated with both an antecedent and a consequent expression that must have the same domain and range concepts. Regarding the 'expression' concept, it must first be noticed that any particular expression may well be associated with several rules. Second, an expression can involve many relations, or concepts, or even constants, which is why we depict an non-specific line from 'expression' to the 'model' area in the metamodel diagram. Finally, it must be realized that expressions in general will constitute derived relations. That is, except in the special case where the expression equals a relation defined in the Unconstrained Conceptual Model.

In the 'explain' area, explanatory texts are associated to concepts, relations and rules, but no particular restrictions apply. In the 'enforce' area of the metamodel, only a uniqueness restriction applies.

5.3 Example of a Meta-rule

In the 'load' area, referential integrity must again be made to hold. As an example how analysis may be conducted and how our language supports the rule designer in her analysis, let us show how to capture referential integrity in a rule enforcement.

Referential integrity holds that for any tuple in any relation, both of its atoms must be on record:

> IF the tuple T has the domain atom A,
> THEN MUST that tuple T is contained-in
> some relation R which has domain some
> concept C which contains that atom A.

This controlled-language sentence translates to a rule in our language as an Horn-clause implication, with symbol **;** indicating composition of relations:

> **rule** 2-referential-integrity-domain **as**
> [Tuple] has-domain [Atom] **must imply**
> [Tuple] contained-in [Relation] (2)
> **;** [Relation] has-domain [Concept]
> **;** [Concept] ~contained-in [Atom]

The antecedent expression is a relation in the Unconstrained Conceptual Metamodel. Hence, this rule assertion (2) permits us to impose the 'resolve by deletion' enforcement strategy: any tuple referring to an unrecorded atom is immediately deleted. This strategy is known as 'cascading delete' in relational database technology. A similar meta-rule will do for the atoms in the range of the relation.

The rule of integrity must hold at runtime, but still its enforcement strategy can be made to vary. A 'reject' or 'report' strategy may be better in a running business environment. Moreover, at load time the 'resolve by addition' strategy is to be preferred.

This illustrates how there are unresolved issues in the analysis and design of the metamodel calling for further research. Another issue is how to deal with constant values in rule expressions.

6 CONCLUSIONS

We proposed a rule language to capture and express declarative business rules. The language, combining business vocabulary with precise mathematical features, is comprehensible for business users, and precise enough to generate rule-compliant IT applications. We outlined how the language may be employed in design- and runtime interfaces.

6.1 Expressive Power of the Language

We claim that the language has adequate expressive power for rule design and analysis.

Following the ideas of SBVR, our language is founded on business vocabulary, compelling the designer to use the plain business phrases for the relevant terms and facts, a major strength of our language. Declarative, invariant business rules are described in a comprehensible if-then syntax. In our experience, this is a great help for people reading a script. In particular, the 'must' keyword provides an immediate clue of what a rule intends to say, even when complicated expressions are involved.

Our language has a clear, uniform makeup. This, and the simple naming regime make for easy-to-read scripts that are straightforward to interpret by business people, even without supportive IT-tools. Each line in a script starts with an imperative keyword that clearly indicates the focus of that line, underpinning the orthogonality of our language. The reserved keywords of our language are concise and learnable, appealing for both skilled business workers and novice rule designers.

Statements of our language are orthogonal by design. Each statement addresses a single aspect of the business context. The language statements are loosely coupled, but as some statements necessarily depend upon previous ones, complete independence is not possible.

No restrictions apply to the order or sequence of statements. The designer may first specify all aspects of one business feature, or start a model with a few rules in one section of the script and add a section with load statements later, etc. Therefore, no particular design approach is forced upon the designer. Having said that, a strong point of our language is that it does force a designer to consider all business features of the relevant rules, and to capture its aspects in distinct statements. For business users, it makes little difference how the statements in the script are organized, because in theory, users do not browse the script but use a dedicated interface to explore the rule-based design. In practice however, users will probably read it, and even begin to add and amend the script.

Our statements are devoid of typical IT jargon such as primary-keys and attributes, functional dependencies, cascading deletes, etc. Imperative ECA-type rules cannot be formulated in our language, with one exception. The enforce statement variant called 'resolve' does initiate data editing operations in response to a rule violation. This is a digression from the strictly declarative and invariant nature of our language, the consequences of which need to be further researched.

In our opinion, rule design for a business context is a superior approach than the dual approach of creating on the one hand an implementation data model with objects, entities, keys, and an activity model with data transitions and processing features on the other. Business stakeholders have little affinity with such refined IT-models, and lack the ability to validate the correct implementation in computer applications.

6.2 Language Extensions

Several extensions to the language may be considered to enhance usability for stakeholders and compliancy of the implementations. Of course, expressive power and understandability for business users should not be affected.

Support for specialization/generalization relations among concepts is one possible extension. This is somewhat problematic for Relation Algebra theory because an atom might belong to more than one concept for some time, or even switch over time: a person is student at one time, and teacher at another. Specialization/generalization is relatively unimportant in business practice, where models often do not need it or can use a work-around.

Better support for the 'resolve' enforcement strategy is needed. Fundamental research is called for to understand the coordination of rules, and to prevent contradictory enforcement strategies.

Support for the Role-based Access standard (Edward et al., 2011) is also suggested. Instances of a Role concept should be assigned the right to access all contents of certain concepts and relations in the Unconstrained Conceptual Model. It calls for a mix of model and metamodel capabilities, thus extending the ideas exposed in section 5.

A serious shortcoming of Relation Algebra is that it lacks arithmetic and temporal capabilities: it cannot express calculations such as 'add 18% VAT' or comparisons like 'if born before 1980'. Support for this kind of rules will greatly enhance usability, provided that the orthogonality of the language and most importantly, the clear and uniform expression of declarative rules in if-then syntax is safeguarded.

6.3 Future Research

We indicate some areas of ongoing research that may improve the applicability of our approach, methods, and tools for business rules design.

Currently, texts available for explanations in the user interface are only static. Ongoing research aims to determine what instructions or explanations in which interfaces are most helpful to achieve high-quality designs (Michels, 2011).

Integration of our language with the typical IT-domain of Semantic Web Rule Language is being researched (Grosof, 2013). The aim is to improve the expressive power without compromising orthogonality of the language and comprehensibility of the if-then syntax of rules.

Interface design is an ongoing area of research. In this paper, we proposed to compose and then compile scripts. But instead of compiling, an interpretative way of working might provide better support for the designer and business stakeholders.

Research is being conducted to develop the reflective meta-modelling approach, its vocabulary and the rules of rule design. The idea is to build a generator from this; a generator that is capable of converting any rule-based design into a fully functional and compliant prototype application.

The vision is that in the future, stakeholders may formulate and validate their own business rules, and do so in a language with enough precision to enable a straightforward implementation in computer applications, without the intervention of IT specialists.

REFERENCES

Akbari I., Yan B.. *Visualizing SWRL Rules*. At ceur-ws.org

Andreescu A., Mircea M., 2014. *Issues and Challenges of Business Rules Modeling in Software Systems for Business Management.* Informatica Economică 18(1)

Bajwa I.S., Lee M.G., Bordbar B., 2011. *SBVR Business Rules Generation from Natural Language Specification.* AAAI Spring Symposium: AI for Business Agility

Beckner M., 2014. *Custom Business Rules. BizTalk 2013 EDI for Health Care*, Apress. p.105-116

Bjekovic M., Proper H.A., 2013. *Challenges of Modelling Landscapes*. BMSD – Business Modeling and Software Design, 3rd Int. Symposium, p.11-22.

Boley H., Paschke A. et al., 2010. *RuleML 1.0: overarching specification of web rules.* LNCS 6403(4) p.162-178

Business Rules Manifesto 2003. Edited RG Ross. At www.businessrulesgroup.org.

Date C., 1981. *Referential integrity*. VLDB.

Decker H., Martinenghi D., 2006. *A relaxed approach to integrity and inconsistency in databases*. Logic for Programming, AI, and Reasoning, Springer.

Dietz J.L.G., 2008. *On the Nature of Business Rules.* Advances in Enterprise Engineering. Springer. 10. p.1-15.

Edward J.C., Timothy R., Rick K., 2011. *Role Engineering: Methods and Standards.* IT Professional. 13: p.54-57.

FBM Working Group, 2011. *Fact Based Modelling.* At www.factbasedmodeling.eu/Data/sites/1/media/FBM1002WD06.pdf.

Fuchs N.E., Kaljurand K., Kuhn T., 2008. *Attempto Controlled English for knowledge representation.* Reasoning Web, Springer p.104-124

Grosof B., Kifer M., 2013. *Rulelog: Syntax and Semantics*. doi=10.1.1.359.9882

Halpin T., 2011. *Fact-Orientation and Conceptual Logic.* 15th IEEE International on Enterprise Distributed Object Computing Conference p.14-19

Hay D., Healy K.A., 2000. *Defining Business Rules ~What Are They Really?* At www.businessrulesgroup.org/first_paper/BRG-whatisBR_3ed.pdf.

Hoare CAR, 1985. *Communicating Sequential Processes*. Prentice-hall Englewood Cliffs

Horrocks I., Patel-Schneider P.F., et al., 2004. *SWRL: A semantic web rule language combining OWL and RuleML.* W3C Member submission

Kardasis P., Loucopoulos P., 2004. *Expressing and organising business rules.* Information and Software Technology 46(11) p.701-718

Lamrani M., El Amrani Y., Ettouhami A., 2013. *On Formalizing Predefined OCL Properties.* International Journal of Computer, Information Science and Engineering 7(1)

Maddux R.D., 2006. Relation algebras. *Studies in Logic and the Foundations of Mathematics.* Elsevier. Vol 150. p. 289-525.

Michels G. et al., 2011. *Ampersand*. Relational and Algebraic Methods in Computer Science. Eds H. de Swart. Springer 6663. p.280-293.

Moody D 2005. *Theoretical and practical issues in evaluating the quality of conceptual models*. Data & Knowledge Engineering 55(3) p.243-276

Object Management Group, 2008. *SBVR: Semantics of Business Vocabulary and Business Rules, Version 1.0.* At doc.omg.org/formal/08-01-02.pdf.

Poulovassilis A., Papamarkos G., Wood P.T., 2006. *Event-condition-action rule languages for the semantic web.* EDBT 2006, Springer. p.855-864

Quartel D., Engelsman W., et al. 2009. *A Goal-oriented requirements modelling language for enterprise architecture.* Enterprise Distributed Object Computing.

Ross R.G., Lam G.S.W., 2011. *Building Business Solutions: Business Analysis with Business Rules.* Business Rules Solutions LLC

RuleSpeak, 2014. *RuleSpeak Sentence Forms, Specifying Natural-Language Business Rules.* At rulespeak.com.

Schön D.A., 1992. *Designing as reflective conversation with the materials of a design situation.* Knowledge-Based Systems 5(1) p.3-14.

Wedemeijer L., 2012. *A comparison of Two Business Rules Engineering Approaches.* BMSD – Business Modeling and Software Design, 2nd Int. Symposium, p.113-121.

Weigand H., van den Heuvel W.J., Hiel M., 2011. *Business policy compliance in service-oriented systems.* Information Systems 36(4) p.791-807.

Modelling Capability and Affordance as Properties of Human/Machine Resource Systems

Vaughan Michell [1] and Ella Roubtsova [2]
[1]*Informatics Research Centre, Henley Business School, University of Reading, U.K.*
[2]*Open University of the Netherlands, Heerlen, the Netherlands*
v.a.michell@reading.ac.uk, Ella.Routsova@ou.nl

Keywords: Capability, Affordance, CPN, Critical Affordance Factor, Affordance Mechanism, Affordance Path, Affordance Chain.

Abstract: Understanding how and why the capability of one set of business resources, its structural arrangements and mechanisms compared to another works can provide competitive advantage in terms of new business processes and product and service development. However, most business models of capability are descriptive and lack formal modelling language to qualitatively and quantifiably compare capabilities, Gibson's theory of affordance, the potential for action, provides a formal basis for a more robust and quantitative model, but most formal affordance models are complex and abstract and lack support for real-world applications. We aim to understand the 'how' and 'why' of business capability, by developing a quantitative and qualitative model that underpins earlier work on Capability-Affordance Modelling – CAM. This paper integrates an affordance based capability model and the formalism of Coloured Petri Nets to develop a simulation model. Using the model, we show how capability depends on the space time path of interacting resources, the mechanism of transition and specific critical affordance factors relating to the values of the variables for resources, people and physical objects. We show how the model can identify the capabilities of resources to enable the capability to inject a drug and anaesthetise a patient.

1 INTRODUCTION

Capability is complex, with wide variations in meaning and evaluation. Capability can refer to the human action ability to do something, (Prahalad and Hamel, 1990) (Gallouj and Weinstein, 1997). Capability also refers to an object's abilities (Beimborn et al, 2005) and the ability of groups of resources to perform a task (Grant, 1991) via a process (Makadok, 2001). Capability relates both to tangible visible transformations, (eg manufacturing an object) and intangible transformations, eg teaching, where information is transferred and tacit knowledge is created (Michell, 2013). The ability to transform resources is the basis of business competitive advantage, where the product resources have greater monetary value than the input resources and cost of work done.

1.1 Paper Objectives and Layout

Our focus is: modelling the capability of a system of business resources to identify how and why it is able to meet a specific capability goal. Such a model enables comparison/selection of the best system of resources for a specific task (Michell, 2012). It also aids understanding the resource properties and dispositions required for a capability-affordance system to achieve a goal. The paper is in 6 sections. Section 2 Introduces affordance and effectivity to formalise the capability model. Section 3 reviews formal affordance models and their shortcomings in relation to capability affordance modelling. Section 4 develops a proposed model for capability analysis using CPN. Section 5 provides an example application of the model. Section 6/7 discusses the use and benefits of the model and our conclusions.

1.2 Definitions

Table1: Definitions.

Environment E	A business environment *E* comprises a set of *{resources Ri}*. The set of *resources* {Ri} have perceivable features whose value at any point is called a disposition
Agent	An *agent* resource is a resource object that can perceive its own environment through sensors and acts on the environment according to their self-motivations through effectors. (eg human or autonomous machine)
Active/Passive Resources	*Active resources* eg a nurse, are capable of exerting a change of state on other object resources in a transformation (note driving resources must be active resources) and have a disposition q. *Passive resources* require other agents to realise their capability ie they are inert and not capable of their own motion or change of state and have a disposition p (eg a syringe).
Action	An *action* is a discrete physical transformation event between active and passive resources that can change the state of a system of resources in an environment to a desired goal state G.

1.3 Capability

A Capability *Cv* results from transformation interactions between two or more resources that achieve a business goal, typically to increase the business value of the transformed resources with respect to a business client. *Business capability is the potential for action to achieve a goal G via an action/series of actions in a process P resulting from the interaction of 2 or more resources, in a transformation that produces business value for a customer.* (Michell, 2011). For example, resources *R1* and *R2* in state *s1* and *s2* interact in the transformation and produce a new state of the system which matches the goal state requirements *G* and in which *R1/R2* may be different. The resources may be combined into a third resource (an input resource is consumed/combined) or *R1* and *R2* remain, but the physical states or *R1* and *R2* are changed. Capability represents the potential of a system of input resources being able to effect a transformation to meet a goal state G and a corresponding system of output resources. For example, a laboratory technician mixing two drugs with a goal to form a new drug, or two doctors discussing a diagnosis. In both cases energy has been expended and a physical state change has occurred. In case *1*, two drugs have been mixed to create a different drug *R3,* but *R1* the drug mixer remains, but in a different state – having the transformation experience. In case *2* information has been passed between clinicians altering their states, i.e. perceptions and memory (biochemistry/ memory state change) *R1* to *R1'* and *R2* to *R2'*). Both transformations add 'value' to the process; a new higher value drug is formed or a patient diagnosis is understood. For the transformation to occur at least one resource must be 'active' and capable of exerting forces and energy via some form of 'mechanism' to transform the other resource. It may be a human or autonomous machine. Other resources may be passive, e.g. drugs, materials etc or also active – another agent or machine. We seek to identify what are the properties of the interacting resources that enable this capability.

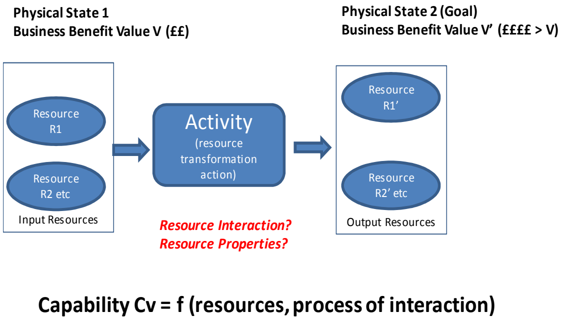


Figure 1: High Level Capability Model.

2 AFFORDANCE / EFFECTIVITY

2.1 Affordance as Environment Ability

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' (Greeno, 1994) Gibson saw affordances as object properties that could be perceived as well as intrinsic properties of the way the object was – its disposition. Affordance represents opportunity for potential action by– visualising what an object can do. Affordance also represents the interaction relationship between the animal and its environment, Gibson's ecological approach identifies action as a result of what the animal or agent can do. Affordances refer to descriptions of (verb-noun) object abilities such as a road is 'walkonable' 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. Affordance is the 'relational' property of the agent environment system that provides the potential for interaction and transformation. It focuses on the possibilities of how the object could be used by the animal or person. However, the animal must also have an ability to use the object and have the correct disposition of properties; otherwise no useful interaction can take place.

2.2 Effectivity as Animal Capability

Shaw (Greeno, 1994) identified that environment ability or disposition, must be complimentary to the animal's disposition and ability. Shaw defined this ability of the animal to compliment the affordance properties of the object as 'effectivity' ie the 'capability of the animal' to use the object in a transformation. Wells (Lenarcic, 2007) suggests effectivity relates to 'the functional state of the animal' and its possible movements. Hence an effectivity; 'can walk', refers to the human ability to meet the goal of walking (Kim et al, 2010). We can think of effectivity as the driving agent ability or the human potential functional characteristics and features that enable them to effect a transformation.

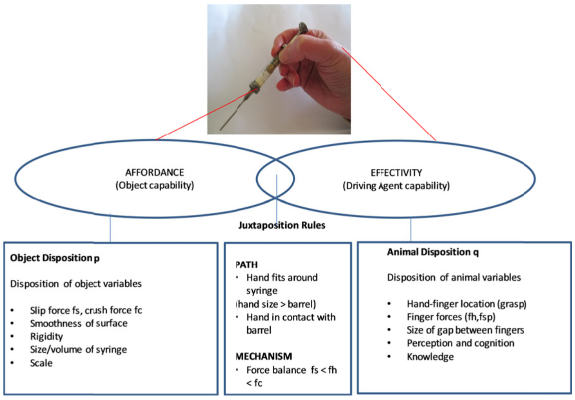

AFFORDANCE
(Object capability)
EFFECTIVITY
(Driving Agent capability)
Juxtaposition Rules
Object Disposition p
Disposition of object variables
• Slip force fs, crush force fc
• Smoothness of surface
• Rigidity
• Size/volume of syringe
• Scale
PATH
• Hand fits around syringe
(hand size > barrel)
• Hand in contact with barrel
MECHANISM
• Force balance fs < fh < fc
Animal Disposition q
Disposition of animal variables
• Hand-finger location (grasp)
• Finger forces (fh,fsp)
• Size of gap between fingers
• Perception and cognition
• Knowledge

Figure 2: Affordance-Effectivity of 'Grasp'.

2.3 The Affordance-effectivity Dual

The disposition of the animal, its effectivity, must compliment the disposition of the object; its affordance, in order for an interaction or transformation to take place. The term affordance-effectivity dual refers to the complement and one cannot exist without the other (Ortmann and Kuhn, 2010). For example to hold an object, such as syringe, a person's hands must fit around the object and the fingers must be in a position to prevent it moving and slipping by applying forces via the fingers, sufficient to lift and hold, but not to crush the object. These are properties of the persons' effectivity in grasping the object. For the object to be held in a hand it must fit inside the hand (volume property), must not be too heavy or slippery, It must also have places that the fingers can grip on to, – properties of the object's affordance. The affordance-effectivity dual refers to the capability of a driving agent to configure (dispose) itself to complement an object's configuration (disposition) to achieve a transformation. The configuration of the agent refers to finger and force location, magnitude, size of hand etc. The configuration of the object refers to its size, weight, slipperiness, functions and features etc. The disposition is a specific set of values of the configuration of object/agent at the point of interaction. A mathematical model is needed to enable quantitative and qualitative definitions for comparison and use of capabilities and their dispositions.

The next section briefly describes mathematical models of affordance and effectivity and their limitations that motivate this paper.

3 AFFORDANCE MODELS

3.1 Turvey et al.

Turvey modelled possible actions, or affordance-effectivity opportunities, as 'prospective controls', to refer to an animal perceiving whether an action is possible (its capability) and its control of the planned action to meet the goal of using the object. Turvey's affordance model related animal properties *Z* and properties of other entities *X* in an environment (Turvey, 1992). The specific animal, *Z* disposition or arrangement *q* enables it to join to *X*, which a complimentary disposition *p*. Turvey has defined the juxtaposition function *j* as the subset of all the possible dispositions *X* and *Z* possess that make the interaction *Wpq* possible. So p (the object disposition) is 'said to be an affordance a thing in the environment *X*' and '*q* is the complementary disposition of the animal *Z*'. At the point of interaction the animal-environment interface possess an interaction property *r*, which is a property of *Wpq* alone.

$$Wpq = j(Xp,Zq) \tag{1}$$

Stoffregen (Stoffregen, 2003) developed the model and identified the need for spatio-temporal contact for affordance to be possible (Lenarcic, 2011). This highlights the need for a space-time path to exist between the interacting agent-object resources that form the basis of our model. Warren and others (Mark, 1987) identified that the geometry of the interacting objects is important in affordance, eg the climb-ability of stairs depended on the ratio of riser height to leg length (Warren, 1984). This suggested classifications of affordance by a dimensionless ratio. However, not all affordances can be reduced to one ratio as Mantel et al (Mantel et al, 2012) observed in their study of action modes, their boundaries and degrees of freedom of interaction. This highlights the need to consider a wider set of critical factors that we refer to in our model. Steedman (Steedman, 2002) used linear dynamic event calculus to identify all the possible potential action paths. However, it does not meet our need for modelling the mechanism of action paths to a specific goal. Brooks (Brooks, 1991) Sahin (Sahin 2007) and others have used affordance extensively to develop ecological behaviour based control in robotics, but this is out of scope of our work, which is focused on human-device work interactions.

3.2 Lenarcic - Situation Theory

Lenarcic combined Barwise' situation theory that models the semantics of situations (individuals, information, time, place) (Barwise and Perry 1980), with Gibson's and Turvey's affordance propositions. Lenarcic's situation theory model relates affordances of a set A of objects in the logic (Lenarcic, 2011):

$$A = \{Aatom, Aset, Astate, Asit, Aaff, Aind\} \qquad (2)$$

Aatom is a set of relevant facts, eg nurse, grasp, hold, syringe etc. Aset is the set of objects. Astate is a set of assertions *{w}* that relates individual people and objects as truth assertions *w = {r, t1…tn, E}* eg *<<in, nurse, room, 1>>*, or 'drug is in the syringe': *<<in, drug, syringe, 1>>*. *Asit*, situations, are sets of relationships between states *{w1,…,wn}*. *Aaff* is a set of affordances as a tuple *{Φ, s, i}, Φ* refers is the action relating to the affordance, s refers to the situation conditions, *i* is the individual capable of affordance, eg *Φ1 <<inject, injection situation, nurse>>* refers to the agent driving the affordance, the action involved and the state conditions. Aind are individuals with their; name, abilities or possible actions eg inject, grasp and their niche or specific action groups (Lenarcic, 2011). An 'enacting function' representing the juxtaposition function, for the affordance to be possible. Lenarcic's model defines a comprehensive algebra for affordances and situations and their semantic relationships. However, the model is complex and unwieldy for more than a few interactions. It is mainly qualitative and hence difficult to compare capabilities or the mechanism of their interaction.

3.3 Affordance Model Developments

Kim et al (Kim et al., 2008) models affordance using situation theory and finite state automata (FSA) models at different levels of detail called grains. A high level grain model represents a plan of action or process and an atomic model of interaction that provides a level of detail within the process that relates to the CAM model. They define a 12 tuple model for Matom:

$$M = (\{X, Z, W\}, \{P, Q, PA\}, Pr, j, \pi, ta, int, tint) \qquad (3)$$

Where the environment is X and human agent Z and W the animal environment system (AES) (Lenarcic, 2011)]. P is the set of affordances, Q the set of effectivities and PA the set of possible actions that can take place. Kim et al include Pr a perceptual predicate function to account for the fact that affordance must be seen and understood in order to use them. Other variables relate to Turvey's juxtaposition function J (the function combining affordance and effectivity) and possible action generation function pi and the goal or target action ta. The tuple concludes with time function for the process level (delta) and the atomic level timing of the affordance-effectivity interaction. Kim et al provide useful examples of the application of the model to a coin in a slot machine and catching a ball. LTL enables notional separation of affordance p and affectivity q (Lenarcic, 2011). However, Kim's 12 sets of variables make it unwieldy in modelling situations where we wish to compare affordances at a higher level of capability, ie several actions. Also it is not easy to model and specify p and q explicitly and intuitively, partly because p and q are related by the juxtaposition function J which is not easily elaborated.

3.4 The Capability-affordance Model

Our model identifies capability as a property of any resource combination animal-animal animal-machine, machine–machine (Michell, 2011). This enables both business capability to be modelled as well as the capability of interacting resources without human intervention eg chemical reactions (necessary as part of industrial processes). Capability requires affordance-effectivity interactions to take place. We take a Gibsonian stance, but unlike Gibson's pure affordance, which relates to possibilities of any resource interactions happening, we are concerned with how and why useful business interactions can happen. Hence goals will be specific to those adding value. Our focus is on determining the conditions and resource specifications for which a specific capability is possible. We illustrate this with the example 'injecting a drug' in a clinical process. Using Gibson and Turvey, we decomposed the affordance-effectivity disposition or possibility for action (Lenarcic, 2011) into (i) a space-time or path disposition and (ii) a mechanism disposition (Michell, 2012). At the point of transition Turvey's juxtaposition function *J* must be represented by both a path and a mechanism, both meet critical affordance factor values that make the state transition possible. The capability of a system of agents and objects is the sum of all the affordance-effectivity interactions within the system. This is equivalent to *W*, the AES- animal in environment system in Kim (Kim et al., 2008). The affordance-effectivity interactions are part of a process where paths represent the what Kim calls 'high grain' interactions and affordance chains represent parts of agents or objects eg syringe components such as the plunger and the barrel interacting.

3.4.1 Affordance Path

The affordance path relates to the space-time affordance-effectivity dual interaction requirements that if the agent and object don't spatially come into contact or a region of influence with each other, affordance won't occur (Lenarcic, 2011). Hence part of the animal disposition *q* and the object disposition *p* conditions must relate to space-time rules regarding the contact/interaction geometry between object and animal. In the syringe example, the syringe position and orientation (*p* variables) must match the hand/finger positions (*q* variables). If the structural spatial arrangement or disposition of the interacting resources do not complement each other, the interaction and capability will not be present, ie if the syringe is too big to fit in the hand or lacks grip and leverage points.

An 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. (adapted from Michell, 2012) At the interaction point between resources, the space time path of animal and object must be the same. Movement and dynamics of the agent in its previous states must be such that it leads to the special agent spatial disposition *q* which matches the special spatial disposition of the object *p* at time *t* of transformation. This becomes a more difficult problem of kinematics when both animal and object are moving and the geometry changes, as in Kim's ball catching example (Kim et al., 2010).

3.4.2 Affordance Mechanism

Having the right spatial disposition alone is not enough. There must be an energy and interaction mechanism to get the resources into contact and to enable the desired cause and effect. For the syringe to be gripped, the hand must exert force on it through the fingers to prevent slipping and crushing. The use of forces in this case is the 'mechanism' or what enables the transformation – to hold the syringe. The affordance transformation mechanism refers to the laws of nature that must hold for the cause and effect interaction between the resources to take place. The most common mechanism in substantive interactions is force, supplied by an animal or machine agent. *The affordance mechanism is the cause and effect transformation at the interface between the two or more interacting resources and its properties that enable the transformation* (adapted from Michell 2013).

Mechanism refers to the behaviour and properties of the energy transfer that drives the transformation eg human energy, chemical, electrical etc. This fits with Gibson's ecological approach. Other mechanisms exist. Chemical mechanisms, enable a substance eg sugar to dissolve in a fluid, if the sugar has appropriate properties ie sufficient surface area and if the sugar's bonds can be broken by a fluid such as water. This represents an object-object transformation between the water and sugar. The mechanism of electric induction depends on the properties of a wire and electromagnetic field and enables an electric current to appear in a wire. This mechanism is necessary for affordance and capability of an electric motor ie a motor affords rotation. Without it the motor has no capability or affordance. Mechanisms are not confined to substantive actions, but include human cognition sense making – or semiosis (Stamper and Liu 1994). The mechanism for the nurse holding the syringe includes the need to perceive the situation (position of the syringe) and the affordance of the object (can the syringe be held – how big/heavy is it, will it fit?). Holding the syringe 'to give an injection' requires different knowledge and skill (repeated affordance experience) than a simple grasp (Andre, 2011) to actualise the affordance–effectivity action of 'inject'. Hence mechanisms should ideally include cognitive resources in terms of 'know what, how and why' that enable the agent to make intelligent decisions to enable the resources to interact. The complete capability model should include perception, cognitive behaviours (Michell, 2013) and capability mechanisms that will affect whether the animal is able to a) perceive and b) understand and bring the resources appropriately together with the right disposition to enable the path and mechanism to effect transformation. For space reasons we only include a brief perception example.

3.4.3 Critical Affordance Factor

In both path and mechanism there is a linked set of critical values of the variables relating to the functions that define the path and mechanism. that enable interaction. These critical affordance values effectively encompass any critical ratios (such as those defined by Warren, 1984) and other factor values that will affect the possibility of the transformation occurring. We define the factors that have critical values attached to them as critical affordance factors, CAF (Michell, 2012). CAFs refer to the values and/or range of values for the disposition of both object and agent to interact ie values of *p* and *q* in Turvey's notation. So in the syringe example they might refer to the range of force (ie critical mechanism values) values to hold the syringe without crushing it and the location/position of fingers within which the syringe can be held (critical path values). This is analogous with Kim's conditions – C (Kim et al., 2008). Identification of critical affordance factors and their ranges are important for both quantitatively comparing existing capabilities and requirements for the capability to exist and in designing new devices and products to meet new capabilities and performance goals.

3.4.4 Affordance Path and Chain

For designed objects, to work, a sequential set of affordances for the interacting components~~,~~ needs to interact in unison in an 'affordance chain' (Michell, 2012). For example, a syringe has a barrel and needle and plunger with a seal that fits inside the barrel. The action of pressing the plunger results in the plunger pushing the air or fluid out of the barrel – eg into a patient. We can say the 'syringe affords injection of a drug'. In an affordance chain (Michell, 2012), the parts are locked together by virtue of their affordance/effectivity properties. An affordance chain also occurs when an agent is holding and manipulating objects, ie the object or tool becomes an extension of the person's hand due to the chain of interactions at object-hand and object-object interfaces. An **affordance chain is a contiguous interaction between affordances acting at the same time.** In contrast the **affordance path refers to a time sequence of related affordances that together produce the conditions for a capability**, eg the capability of injecting a patient.

3.4.5 Resource Properties

Other properties of object and agent are required for affordance-effectivity. For the syringe to be used effectively it needs some kind of scale so that the volume of fluid/drug inside it is known depending on the position of the plunger on the scale. This is an additional necessary resource property of the syringe that enables the affordance-effectivity transformations.

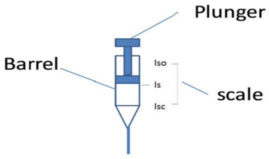


Figure 3: Resource Properties for a Syringe.

3.4.6 Relationship to Other Models

We model the capability affordance model at two levels of Kim's grains, process level for the actions and resources and atomic level for the affordance-effectivity and disposition details (Kim et al., 2008). At the point of affordance-effectivity interaction, the mechanism and path relate to the juxtaposition function. The path represents the space time rules of the juxtaposition function for the affordance-effectivity dual to work. Typically, this will involve the need for objects to be touching, in the same position and specific orientations needed for the transition. Both the object and agent share path spatial conditions. In the syringe example the syringe position and orientation (*p* variables) must match the hand/finger positions (*q* variables). The mechanism represents the action forces (biomechanical, chemical, electrical) etc that enable the transition or change in physical state to occur at the juxtaposition point. Both the object and the agent share mechanism conditions. The critical affordance factors refer to the variables of the path and mechanism and their range values for the affordance-effectivity and hence capability to work. (see Capability-Affordance Model (CAM) below).

4 CAPABILITY SIMULATION

4.1 Model Requirements

Based on the previous discussion, a capability model should enable;

a) modelling of Affordance Paths (process level) at a business process level from initial state to the goal state; representation of affordance path/position and their functions at atomic level, which characterises transitions on the paths.
b) representation of the Mechanism/Force and their functions, at atomic level, which characterises transitions on the paths.
c) modelling of critical affordance factors at atomic level (the values and ranges of their variables) and relationship to path and mechanism functions.
d) The model should show a number of actions at the process and the atomic level that supports a capability without excess complexity. The above are all logical or mathematical constraints and hence a simulation type modeling language with functions and rules is required. Requirements b), c)) rule out traditional BPMN process models and a),d) rules out Kim and Lenarcic's approach. However, Colored Petri Nets (Jensen, 1997) have been widely used to model activities, states and processes. CPN makes the conditions necessary for alternative affordances visible. These conditions are important for capability system analysis. Given our focus on modelling the why and how of capability we propose making them visible as guards of CPN transitions. As we show later the critical affordance factors presented as guards are great tools for simplification of affordance models. Our approach combines CPN with the Capability Affordance Model (CAM) described earlier that allows us to abstract transitions and simplify the model.

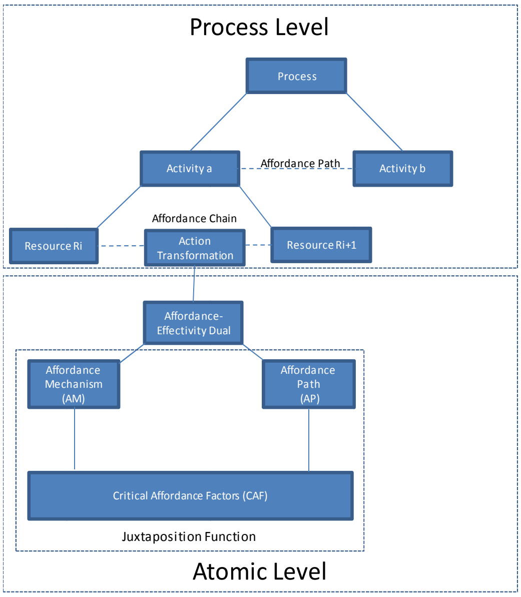


Figure 4: Capability-Affordance Model (CAM).

4.2 CPN and Capability

Coloured Petri Nets (Jensen, 1997) (van der Aalst & Stahl, 2007) possess all the expressive means needed for understanding and possible measurement of the notion of capability. Coloured Petri Nets (CPN) combine advantages of classical Petri Nets (PN) with the expressive power of complex data types "colors "(Jensen, 1997). An initialized non-hierarchical CPN(net) without time stamps is a tuple: *CPN= (C,B,V,P,T),*

- *C* is a finite set of colors (data types), $c \in C$. For example, the colset:
*Syringe=productSyringeName*ForcePlunger* ForceSlip* SyringeScale* SyringeopenScale* (Fig 6.)
represents types of variable needed to describe dispositions (variables) p and q of resources. Colors and variables are defined in declarations. We use data types and variables to represent resource types.
-*P* is a finite set of places, $p1, \dots, pm \in P$, depicted by ellipses (Figure 5). Each place p possesses a bag *bp*.
-*B* is a bag of tokens (values) of colors $c \in$C represented near places. We use tokens to model instances of resources.
-*V* is a set of variables of colors $c \in C$.
-*T* is a finite set of transitions depicted by boxes (Figure 5). Transitions represent actions and are denoted by verbs.
-Each transition is a tuple *t= (I, g,O):*
-*I* is a finite set of input arcs. An input arc is directed from a place *p* to transition *t*. An arc contains an expression of the color of place *p*.
-g is a guard of transition *t*. Each guard is a Boolean function. By default each guard has value true. We use guards to model the critical affordance factors (space time and mechanism) necessary for the affordance-effectivity dual.
-*O* is a finite set of output arcs. An output arc is directed from the transition to a place *p*. An arc is labelled with an expression of a place *Color.*

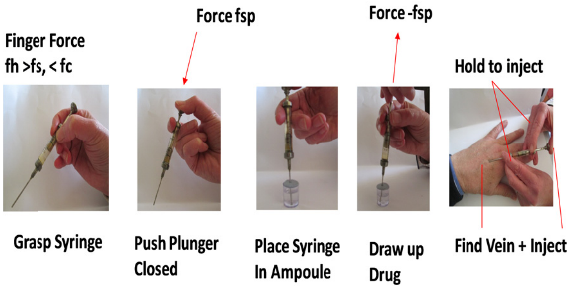


Figure 5: Capability to Inject a Drug Example.

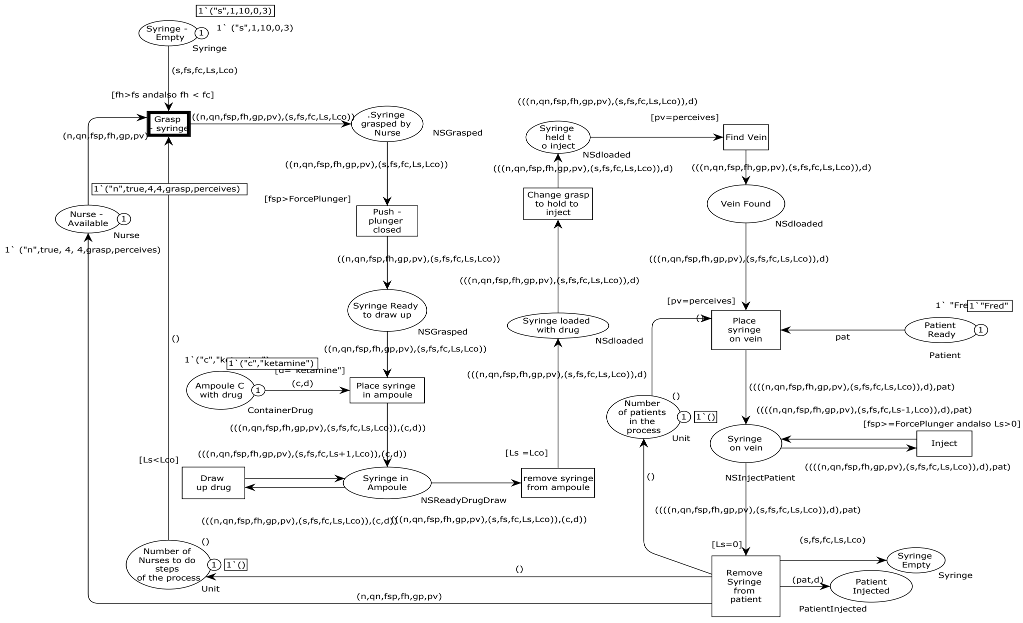


Figure 6: CPN model of Capability - Inject a Drug.

Table 2: CPN-CAM Path and Mechanism Sequence.

Ref	CPN TERMS: P =place, T = transition	PATH *(at point of affordance-effectivity)*	MECHANISM	CRITICAL AFFORDANCE FACTORS *(path/mechanism)*	RESOURCE CONDITIONS
P1	Syringe			fs=1N, fc = 10N,	Empty/clean
T1	**GRASP SYRINGE:** The empty syringe is grasped without slipping, then pushed closed to draw up the drug. The The critical affordance expressions are shown by the guard conditions on 'grasp syringe'. hands fit round syringe, grasp force fs must be greater than slip force (1N) but less than crush force (10N).	hands fit round syringe	grasp forces fh	fh > fs, fh< fc, hands fit round syringe, grasp force fs must be greater than slip force (1 N) < crush force (10N)	Syringe s slip and crush forces, scale ls = 0 and lco =3cm
P2	Nurse				Nurse
P3	Syringe grasped by nurse				NSGrasped - the nurse grasping the syringe
T2	**PUSH PLUNGER CLOSED:** To draw up the drug the syringe plunger is pressed closed by the nurse applying a force fsp > a minimum plunger force. Otherwise the drug cannot enter the syringe.	hand attached to plunger, plunger at end of syringe ls = lsc = 0	plunger force + fsp	fsp1 > forceplunger (min force to move it)	Syringe held in closed position
P4	Syringe ready to draw up (plunger in closed position)			ls =0	NSGrasped - with
P5	Ampoule C with drug			drug type - ketamine	ContainerDrug - an
T3	**PLACE SYRINGE IN AMPOULE:** containing the correct drug. If not in drug no drug will be drawn up (capability failure). The mechanism here is the gasp force holding the syringe and the ampoule - not shown	syringe needle immersed in drug	nurse grasp forces on syringe & ampoule	d=ketamine	ketamine is the correct drug/label for patient
P6	Syringe in Ampoule			ls = 0	NSReadyDrugDraw
T4	**DRAW UP DRUG:** Plunger is pulled back to draw up drug to the correct amount in increments of ls + 1 . Mechanism here is pulling force on the plunger creating a partial vacuum and atmospheric pressure forces the drug into the syringe.	Hand on plunger moved to end of syringe ie ls = Lco = 3	negative plunger force = -fsp	fsp1 > forceplunger (min force to move it), ls = 3	-fsp applied (not shown) ls = ls +1 until ls = 3 on scale
P6'	Syringe in Ampoule			ls = 3	NSReadyDrugDraw
T5	**REMOVE SYRINGE FROM AMPOULE:** The draw up drug continues until ls = 3.= Lco. Incorrect amounts = capability failure / patient not anaesthetised			ls = 3	syringe not in ampoule
P7	Syringe loaded with drug			ls =Lco	NSdloaded
T6	**CHANGE GRASP TO HOLD TO INJECT:** Nurse's finger tip locations/forces adjusted for safe drug injection at correct angle. Failure risks patient injury and not/partly injecting the drug	Grip pattern (position of fingers) = hold	grasp forces fh	fh > fs, fh< fc, GripPattern = hold	Syringe constrained in'hold to inject' position with no slip
P8	Syringe Held to inject			gp = hold	NSdloaded
T7	**FIND VEIN:** A vein on the patient is perceived, based on the nurse' knowledge. Mechanism is nurse' perception/cognition, visual ability (No vein, incorrect site = capability failure)	Visual path: ie nurse can see the patient and the site of injection	perception-cognition mechanism (Pv)	Pv = true	Nurse - has updated knowledge - Vein is found
P9	Vein Found			pv = perceives	NSdloaded
P10	Patient Ready			pat = Fred	Nurse sees the vein
T8	**PUSH SYRINGE IN VEIN:** at correct angle and position			pv = perceives,pat = fred	Correct patient vs drug
P11	Syringe in Vein			ls >0	NSInjectpatient
T9	**INJECT:** Plunger pushed closed at correct injection site to ensure drug is transferred to the patient, (otherwise no anaesthesia and capability fails)	syringe plunger location ls = 0	fsp	fsp > min, ls > 0	Syringe held in closed position
P11'	Syringe in Vein			ls = 0	NSInjectpatient
T10	**REMOVE SYRINGE...** - Inject contintues until ls = 0 (syringe can be withdrawn)			ls= 0	Syringe not in patient
P12	Syringe Empty			ls= 0	Syringe
P13	Nurse Available				Nurse
P14	Patient Injected				Patient Injected

5 EXAMPLE APPLICATION

5.1 Injecting a Drug

Based on structured interviews conducted at a health trust hospital (Michell, 2012) we model the capability to inject a drug using a syringe. Resources include an active resource; a nurse and patient named Fred and passive object resources; a syringe, ampoule containing a drug (eg Ketamine). The capability to 'inject the drug' depends on a process of actions with the correct disposition of resources to inject the drug. If any actions do not have the correct conditions ie any of the critical affordance factors, path and mechanism are incorrect, there will be no capability. The key actions (See fig. 5) are the nurse grasping the empty syringe and pushing the plunger closed ready to draw up the drug. The nurse places the syringe in a drug container (ampoule) and pulls the plunger to draw up the drug. The nurse holds the syringe in a different way – 'hold to inject' and looks for a vein on the patient. Having perceived the vein the nurse pushes the syringe into the vein at the correct position and angle and then presses the syringe plunger to inject the drug. See Table 2.

5.2 Behaviour of the CP-Net CAM Model

Decomposing this process sequence into actions (CPN transitions labelled T) and situations (places labelled P) enables us to identify the critical state transitions and affordances/effectivities. Figure 6 shows a CPN model of the capability to 'inject the drug.'. The initial state and the goal state of the business process are modelled by places that may contain tokens of given colors. Places are connected via transitions so paths leading from initial states to goal states relate to the capability of the system, ie the CPN simulation reaching the goal state. Tokens - represent instances of business object and agent actions and values for the dispositions of each resource (object or agent) at the point of interaction. Transitions represent the transformation affordance-effectivity interactions. A transition *T* of a CP-net is enabled if places of all its input arcs contain tokens to give values to input expressions of *T,* and the guard value is *true*. The guard values represent the **critical affordance factors**. Eg in *T7,* perceives vein must be true for injection to occur. Each enabled transition *t* can fire. When a transition *t* fires then for each input arc its expression is evaluated by a token from the arc's place. For each output arc its function is calculated using the values of the variables from the input arcs of the transition. The result of the output function is added as a token into the place of the output arc. **Affordance** is represented by properties *p* of the passive resources and environment. Eg Syringe –properties are implied by the token: *(s (name) ,fs (slipforce, fc (crush force), Ls (syringe scale level, Lco (scale zero)).* **Effectivity** is represented by properties *q* of the active resources eg the nurse that acts on the syringe, eg Nurse properties are implied by the token: *(n (name), qn (quality), fsp (plunger force), fh (hand force), gp (grasp type), pv (perceives)).* **Affordance Path** at the transformation point, is represented as a net of transitions from the initial places to the goal state *G*. *G* is represented by the state of resources; patient is injected, syringe is empty, nurse is available. **Affordance Mechanism** is modelled with functions corresponding to guards and functions associated with arcs of transitions, eg the force fh applied by the nurse enable the syringe a) to be held in place to execute the affordance-effectivity and b) a second force *fsp* on the plunger moves it to draw up the drug due to the mechanism of a vacuum created in the syringe. **Affordance Chains** represent the concatenation of resource instances and their disposition variables, we use Cartesian products and a value of token with a product type. Cartesian products relate to the Affordance Chain of agent and component objects (eg syringe plunger etc) needed to enable the affordance-effectivity interaction. For example a nurse *holds* a loaded syringe *(NSDloaded)* or an ampoule containing a drug *(ContainerDrug).* The mechanism, path and critical affordance factors at transitions are shown in table 2.

6 DISCUSSION

The CPN Capability Affordance model provides a precise means of modelling and simulating business resource interactions and their capability properties and quantitative values. The model shows that if no affordance (space time) path to the goal state of 'inject' is possible there is no capability to inject. This is represented by the existence of a complete CPN trace to the end state goal. It also shows that capability to inject depends on the mechanism of forces and perception that relate to real-world interactions and conditions. CPNs are executable. This enables critical affordance factors for forces, locations and positions to be identified and modelled so key actions and required properties of the resources for capability 'to inject a drug' can be identified. For space and complexity reasons not all factors are included. For example; a) the nurse must perceive the drug label on the ampoule and ensure it is matched to her knowledge of what drug should be injected into what patient, b) the patient must be perceived and identified by the nurse as the correct patient.

7 CONCLUSIONS

This paper has shown how capability, affordance and critical affordance factors can be presented in a

CPN model. It shows how capability depends on; a) the existence of a possible path of interaction between the resources (nurse, syringe, ampoule, patient), b) a mechanism of transition (forces and drug interaction in this case), c) specific critical affordance factors relating to the actual value of the affordance and effectivity variables for resources such as people and objects within instances, d) That these variables relate to Gibson's original explanation of affordance disposition and the affordance-effectivity dual relationship. Future work will focus on the detail of a single action and its affordance-effectivity relationship by decomposing this into affordance path, mechanism and affordance factors, including perception and planning as well as control actions.

REFERENCES

Andre 2011: http://www.usernomics.com/AnthonyAndre.PDApresentation.WhatMakesAPrefilledSyringeUsableAndErgonomic.July2011.pdf

Beimborn, D., Martin S.F., Homann U. 2005.

Capability Oriented Modelling of the Firm. *IPSI Conference.*

Brooks R.A.. *New approaches to robotics*. Science, 253(5025):1227, 1991.

Gallouj, F., Weinstein O., 1997. Innovation in services. Research Policy, Vol 26, Issues 4-5.

Gibson, J.: The Ecological Approach to Visual Perception. Houghton Mifflin Company, Boston (1979)*California Management Review.*

Greeno J.G. Gibson's affordances. Psychological Review, 101(2):336- 342,1994..

Jensen. K., *Coloured Petri Nets.Springer.*1997

Kim, N., Thiruvengada, H., Rothrock, L., Wysk, R. A., & Shin, D., (2008). Modelling of affordances in human-involved complex systems using finite state automata (FSA). *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Humans and Systems*.

Kim, N. H., (2010). The Formalism of Affordance in Human-machine Cooperative Systems Using Finite State Automata (FSA) (Doctoral dissertation, The Pennsylvania State University).

Lenarčič, A. (2011). Formalizing affordances in situation theory (Unpublished master's thesis).Brock University, Department of Computer Science, Ontario, Canada.

Lenarčič, A., & Winter, M. (2013). Affordances in Situation Theory. Ecological Psychology, 25(2), 155-181.

Makadok, R. 2001. Towards a synthesis of the resource based dynamic capability views of rent creation. *Strategic Management Journal*.22: 387–401.

Mantel, B., Hoppenot, P., & Colle, E. (2012). Perceiving for Acting With Teleoperated Robots: Ecological Principles to Human–Robot Interaction Design. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, *42*(6), 1460-1475.

Mark, L. (1987). Eye height-scale information about affordances: A study of sitting and stair climbing. *Journal of Experimental Psychology: Human Perception and Performance, 13,* 360–370.

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.A. (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.A. Cognition capabilities and the capability-affordance model. Business Modelling and Software Design BMSD 13 Noordwijkerhout, Netherlands 8-10 July 2013

Ortmann, J. and Kuhn, W. (2010). Affordances as Qualities. In Galton, A. and Mizoguchi,R., editor, *Formal Ontology in Information Systems Proceedings of the Sixth International Conference (FOIS 2010)*, volume 209 of *Frontiers in ArtificialIntelligence and Applications*, pages 117–130, Amsterdam Berlin Tokyo Washington,DC. IOS Press.

Prahalad, C.K., Hamel G.: The Core Competence of the Corporation. HBR, May- June 2-15(1990)

Sahin E., Cakmak, M., Dogar, M.R. Ugur, E.. Ucoluk G.. To Afford or Not to Afford: A New Formalization of Affordances Toward Affordance-Based Robot Control. Adaptive Behavior, 15(4):447-472, 2007.

Stamper (1994) SE4 Organisational dynamics social norms and information systems

Steedman M.. Formalizing affordance. Proceedings of the 24th Annual Meeting of the Cognitive Science Society, pages 834-839, 2002.

Stoffregen. T .A. Affordances as Properties of the Animal-Environment System. Ecological Psychology, 15(2):115-134, 2003.

Turvey, M.T.: Affordances and Prospective Control: An Outline of the Ontology. Ecological Psychology 4(3), 173–187 (1992)

Van der Aalst W.M.P. and Stahl. C. Modeling Business Processes - A Petri Net-Oriented Approach. The MIT Press, 2011.

Warren, W.H.: Perceiving Affordances: A Visual Guidance of Stair Climbing. Journal of Experimental Psychology: Human Perception and Performance 10(5), 683–703 (1984)

Using UML to Specify Artifact-centric Business Process Models

Montserrat Estañol[1], Anna Queralt[2], Maria-Ribera Sancho[1,2] and Ernest Teniente[1]
[1]*Universitat Politècnica de Catalunya, Jordi Girona 1-3, 08034 Barcelona, Spain*
[2]*Barcelona Supercomputing Center, Jordi Girona 31, 08034 Barcelona, Spain*
{*estanyol, ribera, teniente*}*@essi.upc.edu, anna.queralt@bsc.es*

Keywords: Business Artifacts, BALSA Framework, UML, Business Process Modeling

Abstract: Business process modeling using an artifact-centric approach has raised a significant interest over the last few years. One of the research challenges in this area is looking for different approaches to represent all the dimensions in artifact-centric business process models. Bearing this in mind, the present paper proposes how to specify artifact-centric business process models by means of diagrams based on UML. The advantages of basing our work on UML are many: it is a semi-formal language with a precise semantics; it is widely used and easy to understand; and it provides an artifact-centric specification which incorporates also some aspects of process-awareness.

1 INTRODUCTION

Business process models have been traditionally based on an activity-centric or process-centric perspective, defining how a business process or workflow is supposed to operate, but giving little importance to the information produced or needed by the process (Damaggio et al., 2011).

In contrast, the artifact-centric (or data-centric) approach to process modeling considers data as a key element in the business process specification. Business artifacts, representing the data, model key business-relevant entities which are updated by a set of services that implement business process tasks. The main advantage of artifact-centric business process models over process-centric approaches is that it is possible to perform semantic reasoning on the model. That is, the meaning of the tasks can be formally defined and it is possible to automatically check for any inherent contradictions in their definition. The artifact-centric approach to business process modeling has been successfully applied in practice (Bhattacharya et al., 2007a) and it has been shown to have a great intuitive appeal to business managers and to system developers.

To facilitate the analysis of artifact-centric process models, (Hull, 2008; Bhattacharya et al., 2009) proposed the use of the BALSA (**B**usiness **A**rtifacts, **L**ifecycles, **S**ervices and **A**ssociations) framework. This framework establishes the common ground for artifact-centric business process modeling by defining four different dimensions which, ideally, should be present in any artifact-centric process model:

- **Business Artifacts.** They represent meaningful data for the business, and as such they hold information needed for the process. Each artifact corresponds to a real life entity, and therefore may be related to other business artifacts. A business artifact has an identity and its progress through the workflow can be tracked.
- **Lifecycles.** Lifecycles represent the evolution of an artifact, showing the relevant stages in its life, from the moment it is created until it is destroyed or archived.
- **Services.** They represent tasks (i.e. meaningful units of work) that evolve the artifact in order to achieve the process's goals. They create, update and delete business artifacts.
- **Associations.** Associations represent constraints in the manner how services make changes to artifacts. This means that associations can restrict the order in which services are executed. They may be specified using a procedural specification (e.g. a workflow) or in a declarative way (e.g. condition-action rules).

Notice that, for the remainder of this work, we use the term *service* with the meaning defined above, which differs from its definition in Service Science.

One of the research challenges in artifact-centric business process modeling is to find the "best" model

(Hull, 2008), seeing that none of the existing approaches can handle the number of requirements of business process modeling. After an extensive analysis, we realized that the majority of work that has been done in this area is based on complex mathematical or logical languages to represent the processes. Although they are practical from a formal perspective, these languages are difficult to understand and not intuitive for business people.

For this reason, in this paper we propose a way to represent artifact-centric business process models such that:

- It is a high-level representation.
- It is independent from the final implementation of the business process.
- It represents most of the BALSA dimensions in a graphical, intuitive way.

We do so by mapping each of the dimensions in the BALSA framework to a graphical model, if possible. We will define these models using the UML language as a basis because of its expressiveness, although they could be defined using a different notation. UML is an ISO standard (ISO, 2012) and it is also the *de facto* standard for software modeling. Although initially conceived for software modeling, we have found that it adapts easily to artifact-centric business process modeling, as we can use it to represent both the static structure and the dynamic behavior of the process. Those aspects that cannot be represented graphically using UML will be specified using OCL (Object Constraint Language) (ISO, 2012). Although they could be represented using another language, we use OCL because it integrates naturally with UML and it does not have the ambiguities of natural language.

This paper extends our previous work in (Estañol et al., 2013) by describing in detail the characteristics of the models we use to define artifact-centric business process models using UML as a basis. We also provide a detailed comparison of our proposal to previous work, and we include some process-centric approaches in the review.

The rest of the paper is structured in the following way. Section 2 presents our proposal for artifact-centric business process models using UML. In order to illustrate it, we will use a running example based on a city bicycle rental system. Section 3 analyzes and reviews the related work. Finally, section 4 states the conclusions and describes some further work.

2 ARTIFACT-CENTRIC BUSINESS PROCESS MODELS IN UML

One of the pending challenges in artifact-centric business process modeling is to find the best model to represent each of the dimensions of the BALSA framework (Hull, 2008). This section describes in detail how UML and OCL can be applied to define an artifact-centric process model following this framework. From this process we obtain a platform-independent model (a term we borrow from model-driven architecture to refer to a model that is independent of the final implementation platform) which is detailed enough to show how the artifacts evolve and the effect of services on them.

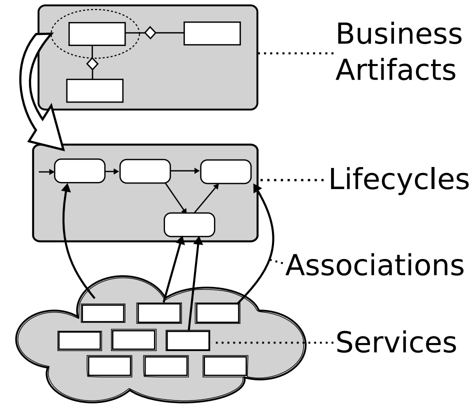


Figure 1: Generic representation of the dimensions in the BALSA framework.

Figure 1, adapted from (Hull, 2008), shows the different BALSA dimensions and how they relate to each other. We start by defining the business artifacts and how they are related to other objects. More specifically, we will determine their attributes, the relationships between them, and their identifiers (for both the artifacts and the objects). Once we have this, for each business artifact, we will define its lifecycle. This should give us an overview of the different events that trigger the transitions between states. After this, it is time to specify the services that make up these events and to establish the conditions or the order of their execution (i.e. the associations).

We will illustrate our approach by means of a real-life example, based on a city bicycle rental system, such as Bicing. Bicing is a service offered by the Barcelona City Council to registered users as an alternative form of transport. Bicycles are docked to stations placed around the city, so that users can take a bicycle form a certain station and return it to a different one. For the purpose of our example, we will consider that users are automatically blacklisted if they do not return a bicycle within three days after taking it.

2.1 Business Artifacts as a Class Diagram

Business artifacts represent the relevant data for the business. A business artifact has an *identity*, which makes it distinguishable from any other artifact, and can be tracked as it progresses through the workflow of the business process execution. It will also have a set of *attributes* to store the data needed for the workflow execution. Business artifacts may be related to other business artifacts and objects. The main difference between a business artifact and an object is that the business artifact has an associated lifecycle showing its evolution, whereas the object does not. In business terms, an artifact represents the explicit knowledge concerning progress toward a business goal at any instant.

The conceptual schema of business artifacts is intended to hold all the information needed to complete business process execution. Business artifacts and objects are similar to domain concepts because they represent real-world entities. As domain concepts can be represented in a class diagram, we will be able to represent business artifacts and objects in the same way. In particular, we will use a class diagram based on UML, which will allow us to represent, in a graphical way, the following elements:

- **Classes**, which correspond to business artifacts or objects.
- **Attributes** for each of the classes, which contain relevant information for each domain concept.
- **Relationships** between classes, showing how they relate to each other.
- **Integrity Constraints**, which establish restrictions over the classes, the attributes or the relationships between them. Constraints can be represented either graphically, in the diagram itself, or textually in OCL or in natural language.

Figure 2 shows the class diagram that corresponds to our bicycle rental system example. The business artifact in our example is *Bicycle*, as it keeps the information of the key elements needed to provide the service: the bicycles themselves, which are identified by their *id*. Apart from the business artifact, there are several objects, such as *AnchorPoint* (identified by *number* and *Station*, according to constraint 3), *User* (identified by *id*) and *Station* (also identified by *id*). For each artifact or object, we will have as many attributes as relevant information we want to keep about it.

The evolution of a business artifact is recorded in the class diagram by means of subclasses. The use of subclasses makes it possible to have exactly the attributes and relationships that are needed according to the artifact's state, preserving at the same time its original identifier and the characteristics that are independent of the artifact's state which are represented in the superclass.

In our example, the business artifact *Bicycle* has four subclasses: *Available*, *InUse*, *Lost* and *Unusable* which keep track of the different stages of the *Bicycle* while containing information which is relevant only for that particular stage. For instance, when a bicycle is *InUse*, it is linked to a certain *User* and has an attribute that stores when it is expected to be returned. These subclasses also require restrictions to ensure that the dates are coherent: we do not show each of the restrictions due to space limitations.

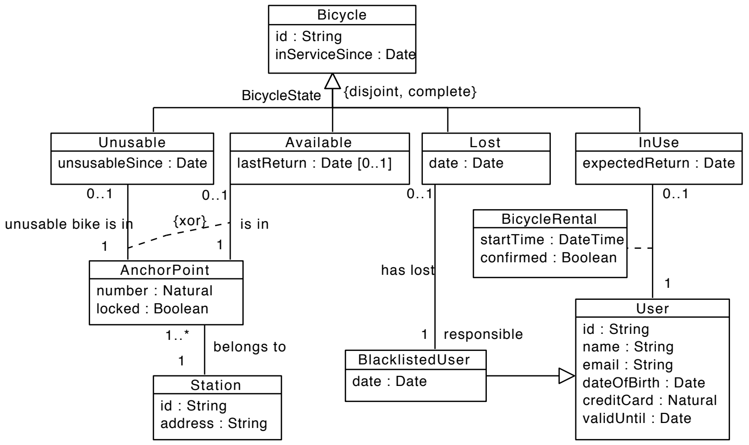
Notice that *BicycleRental* is an association class: it results from the reification of a relationship (i.e. viewing it as class) between two or more classes (Olivé, 2007). Association classes are particularly useful when we want to record additional information about a relationship, as it is the case here. They are identified by the classes that take part in the relationship; in this case, *InUse* and *User* (identified by its *id*).

2.2 Lifecycles as State Machine Diagrams

The lifecycle of a business artifact states the relevant *stages* in the possible evolution of the artifact, from inception to final disposal and archiving, as far as the business process is concerned. It is natural to represent it by using a variant of state machines, where each state of the machine corresponds to a possible stage in the lifecycle of an artifact from the class diagram (Hull, 2008).

Therefore, for each business artifact in our class diagram, we will have the corresponding state machine diagram representing its lifecycle. Taking as a basis UML state machine diagrams, our diagrams consist of:

- **States.** They represent the stages in the evolution of a class or business artifact. Therefore, a state machine diagram will have as many states as relevant subclasses the business artifact has in the class diagram.
- **Transitions.** Transitions, represented as arrows, connect the different states (i.e. stages of the busi-



1. *Bicycles* are identified by their *id*: `context Bicycle inv: Bicycle.allInstances()-> isUnique(id)`
2. *Users* and *Stations* are identified by their respective *id*: `See restriction 1.`
3. A *Station* cannot have two *AnchorPoints* with the same number:
   `context Station inv: self.anchorPoint->isUnique(number)`
4. A *BlacklistedUser* cannot have any *BicycleRentals*: `context BlacklistedUser inv: self.bicycleRental->isEmpty()`
5. Dates should be coherent: `Several OCL constraints would be required.`

Figure 2: Class diagram showing the business artifacts as classes with the corresponding integrity constraints.

ness artifact) in the state machine diagram. Transitions may have the following elements:

- **Guard Constraints.** They are conditions which must be true to trigger a transition between two different states. However, if there is an event in the transition, the event must take place and the condition must be true in order for the transition to be triggered.
- **Events**: Events represent noteworthy occurrences that may trigger a transition between states. If there is no condition, the transition will be triggered when the event occurs; otherwise, the condition must also be true when the event takes place. We will distinguish between two types of event:
  - ∗ **Internal Events.** They correspond to conditions stated over the content of the business artifacts or objects, or over time, and may cause the execution of actions without requiring the user's intervention. They are referred to as change or time events in (Olivé, 2007).
  - ∗ **External Events.** They are explicitly requested by the customer of the business process and their behavior is specified by means of a set of associated services. Therefore, *external events are non-atomic*. They roughly correspond to call or signal events in (Olivé, 2007).

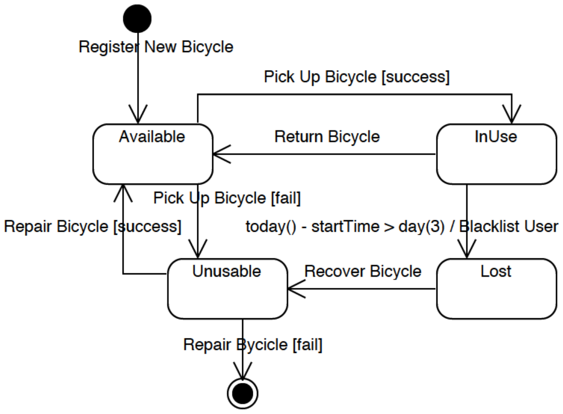


Figure 3: State machine diagram for *Bicycle*.

These events may be accompanied by an **event-dependent condition**. These conditions are placed after the event as they indicate how said event must end in order for the transition to be triggered.

- **Actions.** Actions are automatically executed when the transition fires.

The state machine diagram for the key artifact *Bicycle* in our example is shown in Figure 3. Notice that each subclass of *Bicycle* in the class diagram in Figure 2 now corresponds to a stage in the state machine diagram.

A *Bicycle* is created when it is registered in the system. Initially it is in state *Available*. From the *Available* state, a bicycle may become *InUse*, if a *User* picks it up successfully, or *Unusable*, if he does not (the bicycle may be broken or damaged). Notice that both transitions are triggered by external event *Pick Up Bicycle*, and the final state depends on the event-dependent condition of this event. From state *InUse*, a bicycle will become *Available* again when the user returns it.

When a bicycle is *Unusable*, it needs to be repaired (*Repair Bicycle*) in order to become available again. There are two possible outcomes. If the bicycle is repaired successfully (event-dependent condition *success*) it changes its state to *Available*. If, on the other hand, it is beyond repair, the bicycle is destroyed.

Finally, notice that if a bicycle is not returned for three days after it is picked up (internal event *today()* $-$ *startTime* $>$ *day(3)*), action *Blacklist User* is executed and the bicycle changes its state to *Lost*. If at any time the bicycle is recovered (*Recover Bicycle*), then it changes to state *Unusable*, as it will have to be revised and repaired before it can be used again.

We would like to point out that this diagram does not follow exactly the standard described in (ISO, 2012). This is due to the fact that our state machine diagram has event-dependent conditions, which we use to determine whether the event ends successfully or not. In traditional UML state machine diagrams, events are atomic and there is no need for such conditions.

In addition to this, we also allow having more than one outgoing transition from the initial node. This is useful when the artifact can be created in different ways. Alternatively, this situation could be represented using one outgoing transition from the initial node, leading to a state called *InitialState*. From this state, we could have the outgoing transitions that start from the initial node and leave the rest of the state machine diagram as it is. However, representing the lifecycles in this way does not contribute any relevant information and adds complexity to the final diagram.

2.3 Associations as Activity Diagrams

As we have just mentioned, external events and actions in a state machine diagram are non-atomic and consist of a set of services. Consequently, for each external event and action in a state machine diagram we need to specify the services it consists of and the associations between them. Associations in the BALSA framework establish the conditions under which services may execute.

We propose to specify associations using UML activity diagrams as a basis. Therefore, we will define one such diagram for each external event and action in a state machine diagram. Then, for the state machine diagram in Figure 3, we would have an activity diagram for the following external events: *Register New Bicycle*, *Pick Up Bicycle*, *Return Bicycle*, *Repair Bicycle* and *Recover Bicycle*. We would also have an activity diagram for action *Blacklist User*.

The main elements of activity diagrams are the following:

- **Nodes.** We will distinguish between two types of nodes:
  - **Tasks.**[1]: Tasks represent the work that has to be done. In most cases, each task corresponds to a service (as defined in the BALSA framework). However, tasks with a rake-like symbol are further decomposed in another activity diagram. This is particularly useful to reuse certain behaviors. It is the responsibility of the business process modeler to decide how many tasks and services are needed to specify the behavior of an external event or action. Understandability of the diagrams defined is the criterion that should guide this decision.
  - **Control Nodes.** They are used to manage the flow. Examples of control nodes are decision and merge nodes. Decision nodes indicate that only one of the subsequent paths will be taken according to a certain condition. Merge nodes, on the other hand, indicate that the execution of the activity diagram continues after one of its incoming edges carries information. There are other types of control nodes but we will focus on these two.
- **Edges.** Edges connect the nodes among them and establish the order for the execution of the tasks.

In addition to these, we will also use swimlanes, on the one hand, and notes stereotyped as *Participant*, on the other hand. Although not strictly necessary, swimlanes will provide additional information by indicating the main business artifact or object that is involved in each of the tasks and whether the corresponding task deals with material or informational resources. Material tasks deal with physical elements and are represented using stereotype *material* in the swimlane or in the task itself. Informational tasks, on the other hand, deal with the representation of the

[1] Tasks are actually called *actions* in the terminology of UML activity diagrams. However, in order to avoid confusion with actions in state machine diagrams, we will refer to them as tasks throughout this document.

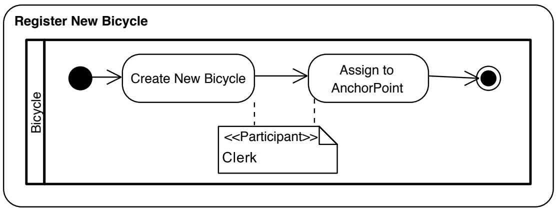


Figure 4: Activity diagram for *Register New Bicycle*.

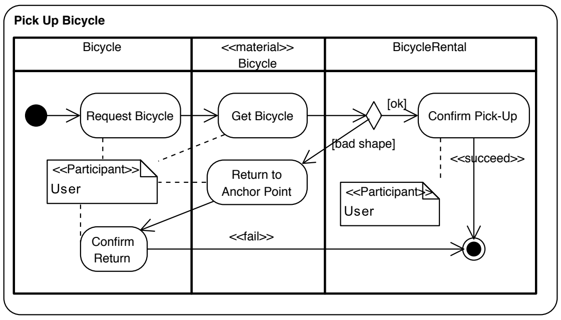


Figure 5: Activity diagram for *Pick Up Bicycle*.

real-life artifact or object. They are shown in the activity diagram as tasks without stereotypes in the swimlane. This distinction is important, because we will only be able to specify services that deal with informational resources and not material ones.

Moreover, we will use notes with stereotype *Participant* to show the role/s of the people who carry out a particular task.

Figure 4 shows the activity diagram of *Register New Bicycle*. First of all, the clerk provides the information of the new bicycle and, after this, he assigns it to an anchor point. The main artifact involved in both services is the *Bicycle*. In this particular case, both tasks are atomic and deal with information and not material resources.

Figure 5 represents the activity diagram of *Pick Up Bicycle*. The user first requests a bicycle to the system and then he physically picks it up from its anchor point. If the bicycle is not in good shape, he returns it to an anchor point and then he confirms the return. Notice that in this case the activity diagram ends in failure. On the other hand, if the bicycle is usable, he takes it with him and confirms the return. Then the activity diagram ends successfully. It is important to make this distinction between success and failure as depending on the result of the activity diagram, the bicycle will change to state *InUse* or *Unusable*, as shown in Figure 3.

In this particular diagram, there are two tasks or services that deal with material resources: *Get Bicycle* and *Return to Anchor Point*, which correspond to physically getting the bicycle from its anchor point and placing it on the anchor point, respectively.

Notice that the approach that we follow to specify associations is *procedural* in the sense that the activity diagrams define a clear order for the execution of services. In contrast, most artifact-centric proposals follow a *declarative* approach where associations define the conditions under which services *may* be executed. Our use of activity diagrams to specify associations brings it closer to process-centric methodologies while keeping its artifact-centric nature.

2.4 Services as Operation Contracts

In the context of the BALSA framework, a service is a unit of work meaningful to the whole business process. Services create, update and delete business artifacts. In turn, this may make artifacts evolve to a new stage meaningful from the business perspective. In our approach, services correspond to the informational and atomic tasks in the activity diagrams.

UML does not have a way to represent services graphically. Instead, they are defined by an operation contract which states what the service does. Operation contracts consist of a set of input parameters and output parameters, a precondition and a postcondition. Both input and output parameters can be classes (i.e. business artifacts or objects) or simple types (e.g. integers, strings, etc.). A precondition states the conditions that must be true before invoking the operation and refers to the values of attributes at the time when the service is called. The postcondition indicates the state of the business artifacts or objects after the execution of the operation. Those artifacts and objects that do not appear in the postcondition keep their state from before the execution of the operation. To avoid redundancies with the integrity constraints of the class diagram we assume a strict interpretation of operation contracts (Queralt and Teniente, 2006).

These operation contracts might be specified by means of different languages, such as natural language. However, the resulting specification would be ambiguous since it would allow for multiple interpretations. Therefore, we propose to specify services using operation contracts defined in OCL. As we have already mentioned, OCL is a formal language that avoids ambiguities. Moreover, it is declarative, which means that it does not indicate *how* things should be done, but rather *what* should be done.

We have shown in Figure 4 that the external event *Register New Bicycle* is performed by means of two services (i.e. tasks): *Create New Bicycle* and *Assign to Anchor Point*, which are informational and atomic. Therefore, we need to specify an operation contract for each such service.

Listing 1: Code for action *CreateNewBicycle*.

```
action CreateNewBicycle (bId: String): Bicycle
localPre: -
localPost:
 Bicycle.allInstances()->exists(b |  b.oclIsNew()
      and b.id=bId and b.oclIsTypeOf(Available)
      and b.inServiceSince = today() and result=b)
```

Listing 1 shows the operation contract of service *Create New Bicycle*. It has as input parameter the *id* of the *Bicycle* that we are adding to the system. The service creates a new bicycle in state *Available* with the given *id*.

Listing 2: Code for action *AssignToAnchorPoint*.

```
action AssignToAnchorPoint (b: Bicycle, apNumber:
     Natural, stationId: String)
localPre: -
localPost:
 b.oclAsType(Available).anchorPoint =
      AnchorPoint().allInstances()->select(ap |
      ap.number=apNumber and
      ap.station.id=stationId)
```

Listing 2 shows the OCL code for *AssignToAnchorPoint*. Given a *Bicycle* and the identifiers of an AnchorPoint (*AnchorPoint number* and *Station id*) as input parameters, it assigns the bicycle to the corresponding *AnchorPoint*. Notice that both the operation contracts in Listings 1 and 2 have no preconditions in order to avoid redundancies. For instance, Listing 2 does not check whether there exists an *AnchorPoint* with the given identifiers.

3 RELATED WORK

This section identifies and examines different alternatives to represent the four dimensions in the BALSA framework. We will mainly focus on artifact-centric approaches, although we will also give an overview of process-centric approaches.

3.1 Artifact-centric Approaches

Although not all researchers who work with artifact-centric business process models specifically use the BALSA framework, it is not difficult to establish a correspondence between their system of representation and the dimensions in the framework. Therefore, in order to facilitate the analysis of the existing alternatives, the subsection has been structured according to these dimensions.

Business Artifacts. Business artifacts, sometimes referred to as business entities, are represented in different ways in the literature. Various authors use database schemas (Bagheri Hariri et al., 2011; Bagheri Hariri et al., 2013; Belardinelli et al., 2011; Cangialosi et al., 2010). A similar representation is proposed in (Bhattacharya et al., 2007b; Damaggio et al., 2012; Fritz et al., 2009; Nigam and Caswell, 2003) where artifacts consist of a set of attributes or variables. Another alternative is to add an ontology represented by means of description logics on top of a relational database (Calvanese et al., 2012). None of these representations is graphical; therefore, this makes them more intricate and more difficult to understand at a glance. For instance, it is complex to see the relationships between the different artifacts.

On the other hand, (Bhattacharya et al., 2009; Fahland et al., 2011; Hull et al., 2011b; Hull et al., 2011a; Damaggio et al., 2013; Lohmann and Wolf, 2010) use different graphical and formal representations for business artifacts. (Lohmann and Wolf, 2010) chooses to represent artifacts as state machine diagrams defined by Petri nets. However, little attention is given to the way the attributes are represented, as the paper is focused on adding the concepts of agent and location to artifact-centric process models. Conversely, (Hull et al., 2011b; Damaggio et al., 2013; Hull et al., 2011a) represent the business artifact and its lifecycle in one model, GSM, that includes the artifact's attributes. The drawback of using GSM is that the relationships between artifacts are not made explicit. Another alternative is using an Entity-Relationship model as it is done in (Bhattacharya et al., 2009). (Fahland et al., 2011) represents the data model of their example by means of an UML class diagram. Both (Fahland et al., 2011; Bhattacharya et al., 2009) use a graphical and formal notation to represent the artifacts' attributes and their relationships, making them easier to understand.

Lifecycles. When it comes to defining the lifecycle of business artifacts, there are two main alternatives: existing approaches either offer an explicit representation of the evolution of the artifact or it is implicit. The lifecycle may be implicitly represented by dynamic constraints expressed in logic (Bagheri Hariri et al., 2011) or the actions that act upon artifacts (Belardinelli et al., 2011; Cangialosi et al., 2010; Bagheri Hariri et al., 2013; Calvanese et al., 2012).

On the other hand, explicit representations are normally based on a state machine diagram. An example is (Bhattacharya et al., 2009). State machine diagrams provide a clear representation of the evolution of an artifact, showing the different stages in its lifecycle and how each stage is reached.

A similar alternative to state machine diagrams is GSM, first described in (Hull et al., 2011b) and further studied and formalized in (Damaggio et al., 2013; Hull et al., 2011a). Like state machine diagrams, GSM allows representing graphically guards and stages. However, it adds the concept of milestone to represent conditions that determine the closing of a state. On the other hand, the sequencing of stages in the lifecycle of the artifact is determined by guard conditions instead of edges connecting the stages, making it much less visual and straighforward. Nevertheless, it is possible to use edges as a *macro*.

Another formal approach is the one used in (Fahland et al., 2011; Lohmann and Wolf, 2010), where lifecycles are represented in variants of Petri nets. A similar alternative is proposed in (Kucukoguz and Su, 2011), where ArtiNets (similar to Petri nets) and DecSerFlow, a declarative language, are used to represent the lifecycle of an artifact and its constraints. These Petri net variants offer an understandable graphical representation that is also formal.

Finally, another possibility is to use a variable in the artifact which stores its state (Damaggio et al., 2012; Bhattacharya et al., 2007b). Although it is an explicit representation, it only shows the current state of the artifact, instead of showing how it will evolve from one stage to the next. Therefore, it is a poorer form of representation in contrast to state machine diagrams, variants of Petri nets or GSM.

Associations. Associations are represented in different ways depending on the approach of the paper. Some authors opt for using condition-action rules defined in logic (Bagheri Hariri et al., 2011; Bagheri Hariri et al., 2013; Cangialosi et al., 2010; Calvanese et al., 2012). (Bhattacharya et al., 2007b) calls these conditions business rules; they should not be confused with business rules in (Damaggio et al., 2012), which are conditions that are superimposed in the already existing ones. In (Damaggio et al., 2012), preconditions determine the execution of the actions; therefore, they act as associations. As in the case of services, associations defined in logic have the advantage of being formal and unambiguous; on the other hand, they will be difficult to understand for the people involved in the business process. In addition to this, the fact that they are not represented in a graphical way makes it more difficult to know, at a glance, the order in which the services will or can be executed.

Likewise, (Bhattacharya et al., 2009) uses event-condition-action rules, but they are defined in natural language. As we have already mentioned, using natural language makes them easier to understand, but they are not formal and may have ambiguities. Moreover, like in the case of associations defined in logic, it is difficult to know, at a glance, the order in which services may be executed, as it is not a graphical representation.

Alternatively, (Fahland et al., 2011) uses what the authors call *channels* to define the connections between *proclets*. A proclet is a labeled Petri net with ports that describes the internal lifecycle of an artifact. Another option is DecSerFlow, that allows specifying restrictions on the sequencing of services, and it is used in (Kucukoguz and Su, 2011). It is a language grounded on temporal logic but also includes a graphical representation. Therefore, both alternatives are graphical and formal.

On the other hand, (Nigam and Caswell, 2003; Bhattacharya et al., 2007a) opt for a graphical representation using flowcharts. The advantage of using flowcharts is that they show, in a graphical and straightforward way, the order in which services will be executed, making them easy to understand. However, they do not use a particular language to define the flow and little attention is given to the way of creating the flowchart.

Services. Services are also referred to as tasks or actions in the literature. Despite the different terminology, in general they are described by using pre and postconditions (also called effects). Different variants of logic are used in (Bagheri Hariri et al., 2011; Bagheri Hariri et al., 2013; Damaggio et al., 2012; Bhattacharya et al., 2007b; Fritz et al., 2009; Belardinelli et al., 2011; Calvanese et al., 2012) for this purpose. The same idea is followed in (Cangialosi et al., 2010; Damaggio et al., 2013) but omitting the preconditions. Using logic means that the definition of services is precise, formal and unambiguous, but it has the problem of being difficult to understand for the people involved in the process.

Conversely, (Bhattacharya et al., 2009) uses natural language to specify pre and postconditions. In contrast to logic, natural language is easy to understand, but it is an informal description of services: this implies that the service definition may be ambiguous and error-prone.

3.2 Process-centric Approaches

It is precisely because process-centric approaches focus on the order of the activities that have to be carried out, that most alternatives only represent one of the elements of the BALSA framework, the associations. As (Weske, 2007) shows, there are many different languages to represent business processes from a process-centric perspective such as BPMN, UML activity diagrams, Workflow nets or YAWL. Although some of these languages have the ability to represent the data needed in the flow (even if in a limited way), process-centric proposals focus on the ordering of the activities or tasks and do not deal with the data.

However, in some cases they do take data into consideration. For instance, (Trcka et al., 2009) represents the associations between services in a WFD-net (WorkFlow nets annotated with Data). The annotation of tasks in the WFD-Net can be seen as a rudimentary way of specifying services, as it allows indicating which data is created, read and written. Similarly, (Ly et al., 2006) uses WSM nets which represent both the control flow and the data flow, although the data flow is limited to read and write dependencies between activities and data. In like manner, (Liu et al., 2007) represents associations in an operational model, which shows tasks (or services) as nodes connected using arrows or edges. The operational model also shows the transfer of artifacts between tasks by indicating them over the edges. However, details of artifacts are not shown.

3.3 Summary of Related Work

As we have seen, none of the approaches that we have analyzed uses the same language to represent all the elements in the BALSA framework, and many times the chosen representation is not graphical, making the models more difficult to grasp. This is aggravated by the fact that, in some cases, the chosen language is a variant of logic which, although formal, makes it next to impossible for business people to comprehend. The use of natural language also has its drawbacks: while it may be easily understood, it may lead to ambiguities and errors.

4 CONCLUSIONS

This paper shows a way to model all of the BALSA dimensions in an artifact-centric business process model using several diagrams based on UML. The use of an activity diagram to represent the associations between services brings it closer to process-centric methodologies and makes it easier to understand than condition-action rules defined in logic or even natural language.

In contrast to other approaches, using diagrams based on UML and OCL as a basis to represent *all* of the dimensions of the model integrates them naturally. Moreover, UML has several advantages: first of all, it is a graphical, high-level language, and therefore the resulting models are easy to understand by business managers, analysts and developers. Secondly, UML and OCL are standard languages. Last but not least, both UML and OCL can be translated into logic (see for instance (Queralt and Teniente, 2012)) in order to facilitate reasoning on the resulting specification.

Finally, our proposal offers the advantages of a graphical representation, understandable by the users, without losing the capacity of being used for reasoning in the future since it is based on a semi-formal notation that avoids ambiguities. As further work, we would like to study the applicability of our approach by analyzing the input provided by the business stakeholders. We would also like to apply this work to a business process model which has more than one business artifact. Last but not least, we intend to define a way to perform automatic reasoning on our diagrams in order to validate their correctness, appropriateness and their quality before they are put into practice.

ACKNOWLEDGEMENTS

This work has been partially supported by the Ministerio de Ciencia e Innovación under projects TIN2011-24747 and TIN2008-00444, Grupo Consolidado, the FEDER funds and Universitat Politècnica de Catalunya.

REFERENCES

Bagheri Hariri, B., Calvanese, D., De Giacomo, G., De Masellis, R., and Felli, P. (2011). Foundations of relational artifacts verification. In Rinderle-Ma, S., Toumani, F., and Wolf, K., editors, *BPM 2011*, volume 6896 of *LNCS*, pages 379–395. Springer.

Bagheri Hariri, B. et al. (2013). Verification of relational data-centric dynamic systems with external services. In *PODS*, pages 163–174. ACM.

Belardinelli, F., Lomuscio, A., and Patrizi, F. (2011). Verification of deployed artifact systems via data abstraction. In Kappel, G., Maamar, Z., and Nezhad, H. R. M., editors, *ICSOC 2011*, volume 7084 of *LNCS*, pages 142–156. Springer Berlin Heidelberg.

Bhattacharya, K., Caswell, N. S., Kumaran, S., Nigam, A., and Wu, F. Y. (2007a). Artifact-centered operational modeling: lessons from customer engagements. *IBM Syst. J.*, 46(4):703–721.

Bhattacharya, K., Gerede, C., Hull, R., Liu, R., and Su, J. (2007b). Towards formal analysis of artifact-centric business process models. In Alonso, G., Dadam, P., and Rosemann, M., editors, *BPM 2007*, volume 4714 of *LNCS*, pages 288–304. Springer.

Bhattacharya, K., Hull, R., and Su, J. (2009). A Data-Centric Design Methodology for Business Processes. In *Handbook of Research on Business Process Management*, pages 1–28.

Calvanese, D., Giacomo, G. D., Lembo, D., Montali, M., and Santoso, A. (2012). Ontology-based governance of data-aware processes. In Krötzsch, M. and Straccia, U., editors, *RR*, volume 7497 of *LNCS*, pages 25–41. Springer.

Cangialosi, P., Giacomo, G. D., Masellis, R. D., and Rosati, R. (2010). Conjunctive artifact-centric services. In Maglio, P. P., Weske, M., Yang, J., and Fantinato, M., editors, *ICSOC 2010*, volume 6470 of *LNCS*, pages 318–333. Springer.

Damaggio, E., Deutsch, A., Hull, R., and Vianu, V. (2011). Automatic verification of data-centric business processes. In Rinderle-Ma, S., Toumani, F., and Wolf, K., editors, *BPM 2011*, volume 6896, pages 3–16. Springer.

Damaggio, E., Deutsch, A., and Vianu, V. (2012). Artifact systems with data dependencies and arithmetic. *ACM Trans. Database Syst.*, 37(3):22.

Damaggio, E., Hull, R., and Vaculín, R. (2013). On the equivalence of incremental and fixpoint semantics for business artifacts with Guard Stage Milestone lifecycles. *Information Systems*, 38(4):561 – 584. Special section on BPM 2011 conference.

Estañol, M., Queralt, A., Sancho, M. R., and Teniente, E. (2013). Artifact-centric business process models in UML. In La Rosa, M. and Soffer, P., editors, *Business Process Management Workshops 2012*, volume 132 of *LNBIP*, pages 292–303. Springer.

Fahland, D., Leoni, M. D., van Dongen, B. F., and van der Aalst, W. M. P. (2011). Behavioral conformance of artifact-centric process models. In Abramowicz, W., editor, *BIS 2011*, volume 87 of *LNBIP*, pages 37–49. Springer.

Fritz, C., Hull, R., and Su, J. (2009). Automatic construction of simple artifact-based business processes. In Fagin, R., editor, *ICDT 2009*, volume 361, pages 225–238. ACM.

Hull, R. (2008). Artifact-centric business process models: Brief survey of research results and challenges. In Meersman, R. and Tari, Z., editors, *OTM 2008*, volume 5332 of *LNCS*, pages 1152–1163. Springer Berlin / Heidelberg.

Hull, R. et al. (2011a). Business artifacts with guard-stage-milestone lifecycles: managing artifact interactions with conditions and events. In Eyers, D. M., Etzion, O., Gal, A., Zdonik, S. B., and Vincent, P., editors, *DEBS*, pages 51–62. ACM.

Hull, R. et al. (2011b). Introducing the Guard-Stage-Milestone Approach for Specifying Business Entity Lifecycles. In Bravetti, M. and Bultan, T., editors, *WS-FM 2010*, volume 6551 of *LNCS*, pages 1–24.

ISO (2012). ISO/IEC 19505-2:2012 - OMG UML superstructure 2.4.1. Available at: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=52854.

ISO (2012). ISO/IEC 19507:2012 - OMG OCL version 2.3.1. Available at: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=57306.

Kucukoguz, E. and Su, J. (2011). On lifecycle constraints of artifact-centric workflows. In Bravetti, M. and Bultan, T., editors, *WS-FM 2010*, volume 6551 of *LNCS*, pages 71–85. Springer.

Liu, R., Bhattacharya, K., and Wu, F. Y. (2007). Modeling Business Contexture and Behavior Using Business Artifacts. In Krogstie, J., Opdahl, A., and Sindre, G., editors, *CAiSE 2007*, volume 4495 of *LNCS*, pages 324–339. Springer.

Lohmann, N. and Wolf, K. (2010). Artifact-Centric Choreographies. In Maglio, P. P., Weske, M., Yang, J., and Fantinato, M., editors, *ICSOC 2010*, volume 6470 of *LNCS*, pages 32–46. Springer.

Ly, L. T., Rinderle, S., and Dadam, P. (2006). Semantic Correctness in Adaptive Process Management Systems. In Dustdar, S., Fiadeiro, J., and Sheth, A., editors, *BPM 2006*, volume 4102 of *LNCS*, pages 193–208. LNCS.

Nigam, A. and Caswell, N. S. (2003). Business artifacts: an approach to operational specification. *IBM Syst. J.*, 42(3):428–445.

Olivé, A. (2007). *Conceptual Modeling of Information Systems*. Springer, Berlin.

Queralt, A. and Teniente, E. (2006). Specifying the semantics of operation contracts in conceptual modeling. In *Journal on Data Semantics VII*, volume 4244 of *LNCS*, pages 33–56. Springer Berlin / Heidelberg.

Queralt, A. and Teniente, E. (2012). Verification and validation of UML conceptual schemas with OCL constraints. *ACM Trans. Softw. Eng. Methodol.*, 21(2):13.

Trcka, N., Aalst, W. M. P. V. D., and Sidorova, N. (2009). Data-Flow Anti-patterns : Discovering Data-Flow Errors in Workflows. In van Eck, P., Gordijn, J., and Wieringa, R., editors, *CAiSE 2009*, volume 5565 of *LNCS*, pages 425–439.

Weske, M. (2007). *Business Process Management: Concepts, Languages, Architectures*. Springer, Berlin Heidelberg.

Is the Value Concept a Valuable Concept for Information Systems?

Coen Suurmond
RBK Group, Keulenstraat 18, Deventer, The Netherlands
csuurmond@rbk.nl

Keywords: Value, Value Chain, Supply Chain, Lean.

Abstract: The creation of value and value chains are popular terms in business management literature. The question is what is exactly meant by value, and what the usefulness of the concept is for operational information systems. In the paper different uses of the term value will be analysed. Subsequently the creation of value for customers will be analysed from a business point of view. The conclusion is that operational processes deal with specifications and norms, which are themselves translations of a background notion of value for the customer in combination with value for the company itself.

1 INTRODUCTION

VAT. Value Added Tax. This fiscal construct shows in its simplicity the core of two aspects of the concept of value in enterprises: value is a financial concept and value is known after the fact. The principle of taxing added value is very simple: an enterprise pays tax over the difference between the total amount billed by its suppliers and the total amount billed to its customers per period of time. The difference between these two values is assumed to be the added value over this period. The present inventory does not factor into determining the added value.

However, when the same tax authorities tax the profits of an enterprise, the situation is very different. In this context it is of great importance to correctly value the present inventory and all other assets of an enterprise. In speculative trade an enterprise can exclusively buy and stockpile goods during a given period, without selling anything, but still make a profit if the inventory is valued at a higher price than the purchase price + purchasing costs + storage costs (e.g. when the current price of the goods is higher than the purchase price paid, this principle forms the foundation for futures trading).

These two fiscal points of view indicate the two different principles of judging value. In the case of VAT only the value of actual sales matters, and this can be objectively determined (although tax consultants can make a nice profit by creatively using fiscal entities and transnational intercompany constructions to ‘optimise’ VAT taxations). For the determination of profits both tangible and intangible assets have to be valued within the boundaries of a framework of fiscal rules (again, subject to ‘optimisation’ by tax consultants).

In the literature about the creation of value and analysing the value chain in a supply chain the same basis seems to be used as that underlying the fiscal principle of VAT. The value of a good is the sum of the costs and the added value is the difference between costs and revenues. At the same time one would expect this approach to add something new to the time-honoured questions of costs and the optimisation of business processes (increasing the efficiency, producing at the lowest cost per produced unit). A reductive view of value as the margin between sales and costs do not address the value-related issues of the attractiveness of the products for the customers before the sales, and the satisfaction of the customers with the products afterwards. Indeed, customer attractiveness and customer satisfaction is an important issue in e.g. the balanced scorecard approach.

In the last few decennia there has been a lot of attention for the efficient set up and execution of business processes in its entirety in management science. This used to be different (or less explicit), the focus used to be much more on performing individual tasks as efficiently as possible. Think of concepts as economy of scale and specialisation or the model of an assembly line as an ideal image of the production process. The transition from “task thinking” to “process thinking” coincides with the analysis of Japanese success stories (that were in turn based on an analysis of Western success stories!) and with the attention for the added value of a process step and the so-called value chain.

Nowadays, the term "value" is used many times and in many ways in the literature about organisations and business processes. It is after all a "valuable" term, it has a very positive connotation. Few people will campaign against the addition of value in business processes, or against eliminating activities that do not add value. At the same time, this is the weakness of the term: what does someone mean when he speaks about the value of a product or service or about the value chain? And if someone discusses the addition of value, what is it that the value is added to? For whom is the value added? Packaging and labelling of crates do not add value to the product itself, but contribute to the efficiency of logistics. Who are the stakeholders in the value chain?

The main part of this paper will discuss the possible role of the value concept in the design of an enterprise information system. How can the value chain approach help in the modelling of business processes, and what kind of information should the information system provide to monitor the value chain? More specifically, how can the value chain approach help in the daily operational control loops in the primary processes?

A prerequisite for the analysis of the possible contribution of the value chain approach is the analysis of the value concept itself. This will be done in a short discussion of the literature about the value chain. Here, three different scopes will be identified: the strategic scope, the business process management scope, and the operational scope.

2 VALUE CONCEPTS AND VALUE MEANINGS IN LITERATURE

Among its meanings of 'value' the OED lists: (1) "That amount of a commodity, medium of exchange, etc., considered to be an equivalent for something else; a fair or satisfactory equivalent or return" (2) "The material or monetary worth of a thing; the amount of money, goods, etc., for which a thing can be exchanged or traded" (3) "The extent or amount of a specified standard or measure of length, quantity, etc" (4) "valour" (5) "The worth, usefulness, or importance of a thing; relative merit or status according to the estimated desirability or utility of a thing" (OED, 2007). The first two entries relate to the exchange value or more specifically the monetary value of a good, the third entry is the objective meaning of the value according to a specific measurement scale while the final entry relates to the utility of a good.

An example of a very reductive application of the value chain concept is the work of Markus Baum about the value chain in the service industry. He cites the definition of Kaplinky and Morris of the value chain "The *value chain* describes the full range of activities which are required to bring a product or service from conception, through the different phases of production (involving a combination of physical transformation and the input of various producer services), delivery to final consumers, and final disposal after use." (Baum 2013, Kaplinsky and Morris 2002), and subsequently analysis the value chain in consulting firms exclusively in terms of a costing model. The value chain approach is used for the classification of the activities in the consulting company as either value adding (billable) or value enabling (non-billable), without any reference to the value of the service to the customer.

In Western literature the emphasis is on the first two meanings and the analysis of the value chain is primarily concerned with cutting costs to improve the competitive position and the financial results. In Japanese literature the emphasis is much more on the final meanings and the focus is on avoiding waste of any kind to improve the quality of the processes. The latter can in part lead to a direct improvement of the financial results, but will always be beneficial to the continuity of the enterprise.

For further analysis of the concept of value it is useful to distinguish a number of scopes where the value concept is employed. Firstly the strategic scope, where entrepreneurs together with internal and external strategic consultants analyse and design corporate strategies. This is the field of Michael Porter, who devised the concept of the value chain as an instrument for strategic analysis. The unit of analysis is the business unit, and Porter states "differences among competitor value chains are a key source of competitive advantage" and "the value chains of such subsets [variations within a business unit for different items] of a firm are closely related, however, and can only be understood in the context of the business unit chain" (Porter, 1985).

Secondly we have the process scope, where business processes are analysed, modelled, and perhaps reengineered. Remarkably, in the seminal work of Business Process Reengineering no mention is made of the value chain. Here reengineering is defined as “starting over” and rejecting conventional wisdom and to keep processes simple “in order to meet the contemporary demands of quality, service, flexibility, and low cost” (Hammer and Champy, 1993). Later on, in Beyond Reengineering, Hammer discusses an exhaustive classification of all work activities in three types: (1) value adding, (2) value enabling, and (3) waste (Hammer, 1996), but the book contains no further use of the value chain concept. In Value Stream Mapping, Martin and Osterling propagate the value stream approach. The relationship with the value chain concept is not discussed, despite the semantic kinship of the two concepts. Like the value chain, the value stream is about the added value of the business processes. Unlike the strategic focus of the value chain, the value stream focuses on business processes and workflows. A value stream is defined as “the sequence of activities an organisation undertakes to deliver on a customer request” (Martin and Osterling, 2014). Moreover, an extended value stream includes those activities that precede or follow on the actual customer request, and the value-enabling value streams support the delivery of value. In other words: the same classification of work activities as we find with Hammer, and a strong connection to the lean movement as initiated by Womack and Jones.

The third scope is concerned with the application of the value chain concept in business modelling for IT systems, for example in the Business Process Management by Mathias Weske. First, he discusses value chains as “a well known approach in business administration to organize the work that a company conducts to achieve its business goals” (Weske, 2013). In the next chapter, Weske presents a hierarchy of the value system, which is composed of value chains, which in turn are composed of business functions. Instead of the decomposition of high level business functions, this approach decomposes the high level business processes, as represented by the value chain. The business functions themselves are further decomposed in business processes and in the next step the business processes are decomposed in activities.

All scopes have two characteristics in common: (1) process orientation and (2) ambiguity or non-specificity of the value concept. It is clear that activities in business processes should contribute to the adding of value to products either directly (value adding activities) or indirectly (value enabling activities). The meaning of the term value itself, however, is mostly lacking. From the literature it can be derived that the meaning of value as in “exchange value” is always present. In value adding activities the assumption is that the value of labour and other resources flows directly into the products. In value enabling activities the value of labour and other resources is transferred later on to the products. The value of the products is expressed in the price the customer pays for the product, but the intrinsic value of the product is its usability and attractiveness with regard to the consumer.

3 USABILITY OF THE VALUE CONCEPT

3.1 Analysis of Value Enabling Activities

The strategy of a company determines on which markets the company will be present, in which role, and with which products. For the development of a new strategy, or for the evaluation of an existing strategy, Porter’s value chain approach might be used, amongst other approaches. The resulting strategy will be expressed in a number of quantitative targets and a number of high level organisational norms. The latter will reflect the non-quantitative values that the company wants to create for the customer and for all other internal and external stakeholders.

The targets and norms will have to be translated into operational facilities, specifications and procedures. These are value enabling activities in the company. Activities can be differentiated according to their nature and horizon. Bigger investments in production and storage facilities have a horizon of several years; minor changes of the physical layout and changes in the organisation of the business processes have a typical horizon of a month up to a year; small changes in the organisation of resources might have a horizon of several days or weeks. The common factor in all these activities is their orientation on expected volumes of production, and on expected variability and volatility of demand. These activities are bigger or smaller investments in intangible assets, in order to improve the effectiveness and efficiency of the primary processes (just as a reminder: a fixed asset represent a value on the balance sheet of the company). Sound management requires that the presuppositions on which these value enabling activities are based are carefully checked before and after, in that sense creating a control loop to be supported by the enterprise information system.

Apart from availability of the physical facilities and the organisation of the business processes, a further prerequisite for the actual execution of value creating activities is the determination of the range of products, the definition of each product, and the specification of the production processes, the raw materials and the resources. These definitions and specifications will bring forth further operational norms for inbound logistics, operations, and outbound logistics; and a calculation of the cost price. Product development, process development, and the marketing of the products are all value enabling activities. The marketing function (often both in a marketing department and in actual sales) translates the strategy in actual products for actual markets, and specifies the product qualities and possible ('tolerable') cost prices. This is the translation of value for the customer into specifications and norms for production.

Note: obviously, the marketing function often clashes with the production function, and in a healthy company product and process development is an iterative process in which marketing, sales, production, and quality assurance functions are involved. Nevertheless, the main point is that the marketing function translates perceived customer value into product specifications.

3.2 Analysis of Value Creating Activities

In de daily operation, actual products are produced, shipped, and billed to actual customers. The organisation of production might be make to order, make to stock, finish to order, and many other possibilities, but the bottom line is the fulfilment of customer orders. In most situations, the balancing of demand and supply is a major challenge. The simplest model is pure make to order: the primary processes are only processing actual orders, and all resources are made available on demand. However, the normal situation is that primary processes are executed based on expected demand. When the expectation is correct, customer orders can be delivered quickly and the company has minimal stocks. A mismatch of expectation and actual demand results in bigger stocks and later delivery of customer orders. Clearly, elements of customer value are involved here. Shorter lead times are better for the customer, and stocks cost money and erode profits. The ideal situation for the customer is a storage facility nearby, and his demand can be fulfilled immediately (indeed, this involves both the Just-In-Time concept and the Vendor-Managed-Inventory concept). These customer values are formally or informally translated into a kind of service level agreement, where lead times and delivery frequencies are specified. Production planning will seek the balance between (1) fulfilment of customer orders within the specified lead times, (2) minimal stocks; and (3) maximum efficiency. Production itself must be efficient and it must deliver quality (as specified).

Given all the talk about value creating processes, it a strange phenomenon that conventional information systems use concepts such as Demand Management and Sales and Operations Planning (Vollman e.a. 2005). Demand and sales are externally triggered by the market. What is managed and planned by the company is the availability of products for the markets, given the variability and volatility of the markets (Packowski, 2014). Rigid "sales planning" will yield suboptimal responses to market forces and cause less availability of products and higher costs in the supply chain.

3.3 Management and Control of Value Creating Activities

The value creating activities are controlled by two objectives: creating value for the customer, and creating value for the company itself (margin). The direct indicators for the creation of customer value are the checks on the quality of the product, on the quality of the production processes, and on the quality of the delivery processes. The direct indicators for the creation of value for the company itself are production data about inputs and outputs for the various process steps. In the daily execution of the primary processes, the creation of the customer values within agreed and reasonable boundaries have priority, and operational costs come in second place. However, all deviations of production costs from the standards outside a reasonable bandwidth should be explained, and all explanations should be categorised and kept for periodical analysis. Any deviation is an indicator for either a faulty specification or norm, or some unforeseen irregularity in the outside world that forced an adaptation in the primary processes. Small but regular deviations with internal causes might be more important than big and irregular deviations from outside causes. Incidentally, all deviations should be explained, not only the negative ones. When consumption of resources is below the specification, it might indicate that the process was not correctly executed, or data was not correctly captured, or that under certain circumstances fewer resources are needed than specified. In all cases, the deviation should be explained.

Over a longer period of time, however, the value for the company itself has priority in the evaluation of the deviations from the specifications. Analysis of the differences will suggest causes in one or more of the following categories: substandard execution of the primary processes themselves, too many disturbances of the primary processes from the ordering processes within the agreed and reasonable boundaries (caused either externally or internally!) , or too many unforeseen or unallowed disturbances from the ordering processes. The company will act to improve the internal processes, to adapt the specifications and norms, to agree different terms with the customer, or to adapt the ordering processes.

Besides consumption and net output all kinds of waste will have to be measured as well as an integral part of the capturing of production data. The OEE approach forms a good example of this for individual production lines (Hansen, 2001). For non-line processes all kinds of stoppage, rejection, degradation and waste will have to be measured as well. By also recording the cause (no resources available, no removal of output possible, machine defects in the line, insufficient quality of raw materials) and verifying these causes a continuous PDCA cycle will come into effect. For example consider production losses due to rush orders. The first step is specifically writing the additional set up times against this specific rush order. The second step is checking where the rush order originated from: internal laxity (by whom?), or a late order by a customer. When it concerns an incidental rush order by this customer and this falls within the commercial agreements, then there is no problem. When the same customer keeps on placing rush orders, sales will have to take action with this customer. When continuously incidental rush orders by different customers lead to process losses that are not taken into account in the cost price, then action will have to be taken on the production side, on the commercial side, or both. In each of these considerations and actions “value for the customer” plays an essential part by its influence on the price of the product and, especially, by its influence on service levels, but these are background norms. Human considerations and interpretation leads to the establishment of standards for the production, and deviations from these standards will have to be noted and explained. Individual employees might do valuable things for creating value for the customer by dealing with disruptions, at the expense of extra consumption of resources. These disruptions can originate from the customers themselves (last minute changes in orders), from internal departments upstream or downstream, or from suppliers. The more the enterprise is able to deal with disruptions and to dampen their effects, the more the enterprise will contribute to the stability of the processes of its customers, and in this way create value for them.

3.4 Management and Control of Value Enabling Activities

The value enabling activities are instigated by a set of expectations or plans of future events. Management of the value enabling activities requires a continuous check on the actual course of events and on the actual contribution to the creation of value. Signs of possible drift, slow and continuous changes in the business environment, should be observed carefully, because an accumulation of tiny changes under the radar might have a big impact. More important, the value enabling activities should be checked continuously for their contribution to the value creating processes. It means searching for patterns in the deviations in the daily operations, and searching for operational improvement. The deviations might be found either in defective products, processing problems, or meeting customer demands.

Each and every check, however, deals with specifications and norms. Value in itself is not measurable, only its translations in specifications and norms (and an expectation is an implicit, unexpressed, norm). As the creation of value for both the customer and the company itself is the ultimate objective, a major management issue is the validity of the translation itself. A further issue is the way the organisation measures deviations from the specification or norm. A body mass index does not measure the health of an individual, it is just an indicator for health risks. A best before date does not represent the freshness of the food, it represents just an amount of days after production date as an indicator for freshness (under the assumption the product is kept in the right environment).

3.5 Margin Management

Margin management is an instrument for managing both value creating and value enabling activities by monitoring the service levels in combination with the differences between normative and actual consumption of resources. The final outcome of margin management is ultimately expressed in the financial results (normative and actual). The real value however lies in analyses of causes and explanations of deviations. Daily checks and daily questions about anomalies and differences keep the employees alert and provides valuable information to the "value enablers" about the primary processes as executed under pressure of reality (this is an example of elicitation of possible relevant differences between model and reality). Searching for patterns over time can bring trends to the fore that require adaptations in the way the primary processes are set up.

In analysis of deviations both causes and explanations are important. Cause and effect belongs to causal chains, "Y happened because of X". Explanations have a different nature, explanations are about intentions and norms. The organisation of the primary processes and the specifications of products and processes are the cause of chains of events with input and output. A malfunctioning machine is the cause of an amount of rejected finished product. An unreliable forecast causes loss of net production time. A late customer order, however, might be the explanation of a late departure of a truck. Or the first order of a new customer can be the explanation of extra consumption of resources (the salesperson had emphasised that "first time right" was an absolute requirement for this order). To put it differently, explanations, not causes, give information about intended value for a stakeholder in the value chain.

Important is that margin management does not necessary operate on the level of the individual order or job. It operates on the level where production data can be captured meaningfully and reliable, which is often on a higher aggregation level than capturing the consumption of each and every resource per each and every job. For different resources different levels of aggregation might be used. E.g., quantities of raw materials, semi-finished products and finished products are captured per job, or output per job and input per line; labor is captured per department, energy is captured per area, and waste per line. In margin management this does not really matter for the management mechanism itself: just calculate the normative consumption on the aggregation level on which the consumption data are captured, make the comparisons, analyse the deviations, and record causes and explanations. Here again, value created is not the prime focus, but consumption service levels. The creation of value can be an important issue, but only in an explanation of a difference. A refinement of this mechanism is the capturing of losses (compared with the general norm) with cause and/or explanation as an integral part of the primary processes, with checking of the actual differences afterwards.

3.6 Recapitulation

The concept of value has no place in an operational information system. It might be a very useful concept in the determination of the strategy of a company, but it has to be translated into concrete specifications and norms to be of operational use. Management should continuously check the translation of values in specifications and norms, and analyse the patterns in deviations. Operational people do not create value; operational people create products according to specifications and behave according to norms. Customer value and corporate value are the indirect results of the work of operational people.

However, norms and specifications may conflict. In these cases the behaviour of the employees in the business processes is no longer a routine of following rules, but values and priorities have to be weighed. It is here that the values for the customer and the values for the company that lie behind all the specifications and rules come into play. Both in making the judgement and in accounting for the choice afterwards, it is important that the values as perceived by the employee are made explicit and are evaluated. Just because the people in the primary processes have different information and are dealing directly with the products or the customer, they may shed a valuable light on the perception of value.

4 EXAMPLE: RETAILER

As an example case we consider a big retailer that operates nationwide and has a market share over 20%. The main product segments are fresh products with a short shelf life (meat, vegetables, fruit, bread), food products with a longer shelf life, dry grocery products, and non-food products. The operation of such a retailer is composed of the stores, distribution facilities consisting of distribution centres and transport from the distribution centres to the stores, and the supply to the distribution centres.

The category management department determines which products of which brands are carried at what prices, as well as the way in which the products are offered to the customer. The replenishment function can be located at the individual stores (ordering procedure) or it can be centralised (pushing of products to the stores). The grocery and non-food products are always supplied by third parties. Fresh products such as bread or meat can either be supplied by third parties or produced by the retailer itself. Food products with a longer shelf life are usually supplied by third parties.

For the retailer as a company two primary values can be distinguished: the contribution of the different product segment to the financial results, and the importance and reputation of the product segment by the customer. The latter is a contribution to the pull of the retailer to get the customers into the store and as such is also a factor in the revenue of other segments. The former contribution has a direct financial nature and can be determined by the usual means (which have to a certain extent a discretionary component). The second contribution is primarily of a qualitative nature and has financial effects besides. The attractiveness and image of a retailer determine in part whether a customer is going to do his shopping there, thus contributing to the results. The attractiveness to the customer is determined by the presentation and availability of products in the stores and by the quality and price of the products. The responsibility for these aspects is located partly with category management and partly with the replenishment processes.

The replenishment function is supported by a number of logistic processes. Distribution logistics is a trade-off of on the one hand lead times, frequency and size of the deliveries and costs on the other hand. The shorter the lead times, the more frequent delivery can be and the lower the ordering volume, the better the replenishment process can be performed. However, the logistic costs will be lowest when the number of deliveries is as low as possible, when the utilisation of the capacity of the vehicles is as high as possible, and when handling costs in the preparation of the branch order, in order picking, and in transport are as low as possible. In planning the deliveries a longer lead time means more efficient processes in the preparation of the order fulfilment and transport.

The options for the consumer are determined by the number of products on offer and is expressed by the number of different articles, the different article variants (low cost, premium, organic, …) and/or brands (national brands, store brands, no-brands) and the various packaging sizes. The other side of the coin is that sometimes the consumer cannot see the forest for the trees anymore and gets lost among the options. At this point the freedom of choice flips to a necessity of choosing and the diversity on offer can become a negative feature for the consumer. This is why the presentation of products is of such importance in category management: a logical division of products with the right presentation both on the product itself and on the shelves helps the consumer to make a choice; a multitude of overlapping product categories with poor consumer information (such as unclear logo's) will deter customers.

Value for the customer of the retailer means an attractive shop, and full shelves with the right products. Who creates this value? Primarily the shop designer, category management and replenishment. What are the employees doing in the primary processes of the internal and external suppliers of the retailer? They produce the right products, not directly according to the wishes of the consumer, but by producing in accordance with the specifications. Part of the specifications are determined by the end product itself, and are meant to be appreciated by the consumer. Part of the specifications are determined by the logistic processes (both packaging and information issues), and contribute to lower logistic costs and a faster throughput. The first element contributes to value for money for the consumer and/or the profit of the retailer, the second element contributes to availability and freshness of the products, another value for the consumer. Evidently, as any healthy company the retailer strives for profits and market share. Consumer value is a means to an end. Perceived value is boosted by marketing campaigns, attractiveness is enhanced by sport sponsoring. A retailer is no charity institution, and talk about consumer value sometimes is quite hollow. What matters is that the retailer must provide value for money, and translates this abstract concept via category managements and replenishment functions in operational specifications and norms.

5 RELATED WORK

The "value chain" concept is strongly process-oriented, while at the same time the determination of value is strongly product-oriented. In the literature, the value of a product in a business sense is mainly seen as the difference between costs and revenue (and the costing model is considered unproblematic). On the other hand, the literature often discusses the creation of value for the customer. This means that the product has characteristics that make it valuable to the customer. The process approach of the value chain claims to analyse the business processes according to their contribution to the value of the product for the customer. This approach has strong connections to the lean philosophy originating in Japan. However, there is a difference in emphasis: while the value chain analysis is primarily concerned with the contribution to value, the lean approach emphasises the avoidance of waste of any kind. This is an important distinction: the contribution of value looks at the difference between costs and revenues; the lean approach looks at all kinds of waste, regardless of value, in order to arrive at a product with the right characteristics. Both tend to take the usability and attractiveness for the customer for granted.

For the application of the value concept for operational processes two approaches must be considered, namely the supply chain approach and the lean approach. The supply chain approach is in part dealing with the same issues as the value chain concept, and the value chain could be described as encompassing the value chain. The problem with the supply chain, however, is the ostensible denial of the market mechanism. Take the following definition: “Supply Chain Management can be redefined as a strategic channel management of networks of business integrated together through information technologies and empowered to execute superlative, customer-winning value at the lowest cost through the digital, real-time synchronisation of products and services, vital marketplace information, and logistics delivery capabilities with demand priorities” (Ross 2011). A very ambitious and verbose definition, but what does it mean in our real world? Who are the actors, and what is the relationship between the actors? The supposition is that the actors are interdependent companies, acting for the greater good of the optimal supply chain as a whole. But, low level market-oriented questions remain: who orchestrates the supply chain? Which companies take the profit from the surplus value of the supply chain? Power relations between companies matter, and when a weaker company cooperates with a stronger market power, chances are that the weaker company is squeezed to the point of barely surviving (or pushed beyond that point). Co-creation of value is a very nice idea when you are the stronger partner, for the weaker partner it implies the denial of independent value creation for its own good.

The connection of the value chain with the lean approach is much more fruitful. But where the value chain talks about the creation of value, one of the main focal points of the lean approach is the elimination of waste. Another main focal point is the quality of both process and product. These two forces result in the effective and efficient production of the specified products. Provided that the specified products are of value for the customer, the creation of value is unavoidable outcome of the lean process.

6 CONCLUSIONS

The concepts of “value” and “value chain” have been introduced by Porter as an approach to support strategic decision making by the enterprise. Afterwards, these concepts have been applied increasingly to the analysis of business processes, with a strong connection to the ideas of the lean approach. The concepts seem to focus primarily on financial issues, translated to results as the difference between costs and revenues. This inclination to cost aspects has two implications for the usability in an operational context. Firstly, people in operational processes are not primarily engaged with cost as such, they are doing their job in an effective and efficient way. Both the organisation of the processes and the execution of the processes contribute to low costs, but people are accountable for working according to specification and for avoiding waste of resources. The second implication is that the usability of the products is not determined by the processes themselves, but by the product specification. This is the area of product development and marketing, where the issues of offering the right products for the right price for the target markets are addressed.

An important aspect of the value chain is the interaction of the company with the customers. In this context, the primary processes should be able to adapt to some degree of variability and volatility of demand. To which degree is determined by the market strategy of the company, and translated into operational norms for the primary processes. Again, value for the customer is decoupled from the operational processes via operational norms. Operational people are dealing with these norms and not with their own individual perception of value for the customer (unless the company attributes this specifically to some employees).

The value concept is a fuzzy concept, in the sense that the concept has different meanings that are used in a fuzzy way. One core meaning is practically useless as it will not add value to any discussion: value as the margin between cost and revenues. Another core meaning is usability for the customer, and this is an important notion (and has always been so). The value chain seems to be using this notion, but does not clearly say so. In determining the strategy of a company this meaning of value (and value chain) might well be used.

In the analysis of the processes and activities within the company the classification of value creating, value enabling, and other (wasted) activities can provide a useful vocabulary. For the actual activities, however, the value notion is no more than a background notion. The 'usability for the customer' must be translated in specifications and norms, that are the guidelines for the execution of the primary processes. In resolving conflicting norms in the execution of the primary processes the value for the customer (and the value for the company) might be factored in, both in making choices in the process and in defending them afterwards.

REFERENCES

OED, 2007. *Shorter Oxford English Dictionary*, Oxford University Press. Oxford, 6th edition

Baum, B.B., 2013. *Service Business Costing*, Springer Gabler. Wiesbaden.

Kaplinsky, R., Morris, M., 2002. *A Handbook for Value Chain Research*, Institute for Development Studies. Available at http://www.globalvaluechains.org/tools.html

Porter, M.E., 1985. *Competitive Advantage*, Free Press. New York.

Hammer, M., Champy J., 1993. *Reengineering the Corporation*, Nicholas Brealey. London.

Hammer, M. 1996. *Beyond Reengineering*, Harper.New York.

Martin, K., Osterling, M., 2014. *Value Stream Mapping*, McGraw-Hill. New York.

Womack, J.P., Jones, D.T., Roos, D., 1990. *The Machine that Changed the World*, Free Press. New York.

Womack, J.P., Jones, D.T., 1996. *Lean Thinking*, Simeon Schuster. London.

Weske, M., 2007. *Business Process Management*, Springer. Berlin.

Vollmann, T.E., Berry, W.L., Whybark, D.C., Jacobs, F.R., 2005. *Manufacturing Planning & Control Systems for Supply Chain Management*, McGraw-Hill. New York, 5th edition.

Pachlowski,, J., 2014. *Lean Supply Chain Planning*, CRC Press. Boca Raton.

Hansen, R.C., 2001. *Overall Equipment Effectiveness*, Industrial Press. New York.

Fernie, J., Sparks, L., 2014. Retail Logistics: changes and challenges. In: *Logistics and Retail Management*, Kogan Page. London.

Nielsen, A.C., 2006. *Consumer-Centric Category Management*, John Wiley. Hoboken NJ.

Ross, D.F. 2011. *Introduction to Supply Chain Management Technologies*, CRC Press. Boca Raton, 2nd edition

A Test Generator for Model-Based Testing

Ella Roubtsova[1] and Serguei Roubtsov[2]
[1]*Open University of the Netherlands*
[2]*Technical University Eindhoven, The Netherlands*
[1]*Ella.Roubtsova@ou.nl*, [2]*s.roubtsov@tue.nl*

Keywords: Model Based Testing, Test Generator, Case Management Solutions.

Abstract: The idea of software testing by software itself drives the attempts of creation of model based testing approaches. Model based testing relies on models, that encode the desired behaviour of a system, the relevant behaviour of its environment; however, both the static and the behaviour models are not directly applicable for model based testing. The models should be different from implementation and possess the observational consistency allowing preservation behaviour of parts in the behaviour of the whole. The models need to be built into a tool for preparation of the model and data for test generation. This paper proposes to use protocol models for model based testing. A protocol model presents the external behaviour. Unlike other modelling approaches, the Protocol Modelling semantics incorporates data into the model. This property gives us advantages for the choice of the strategy of test generation. The main contribution of the paper is a combination of algorithms for test generating from a protocol model. The preparation of a protocol model for test generation and the results of test generation are illustrated with an example.

1 INTRODUCTION

What is a software test? It is a procedure or a sequence of actions that allows one to prove that the software meets the requirements. A test may come from requirements directly or may be generated (Figure 1) from a model different from the implementation model, as it is suggested in the UML test profile (UML, 2005).

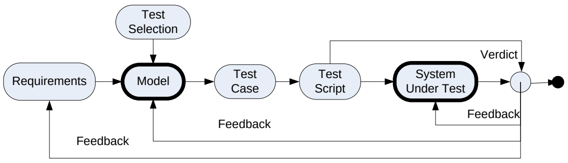


Figure 1: Model-Based Testing.

Development of a dedicated test generation model is often considered by system developers as unnecessary overhead, unless the model provides obvious reduction of testing effort. The choice of a behaviour modelling method with right properties facilitates reduction of testing effort and simplifies further system maintenance.

The right properties of the modelling method depend on the application domain. In this paper, we focus on the domain of Case Management Solutions (CMS), that covers the systems that “coordinate a service request in finance, health, legal, citizen or human resources-related matters, on behalf of a subject such as a customer, a citizen, or an employee” (Davenport and Nohria, 1994; Aalst van der and Weske, 2005). The domain of Case Management Systems (CMSs) has been defined in an OMG document. ”A Case is a proceeding that involves actions taken regarding a subject in a particular situation to achieve a desired outcome... Case management planning is typically concerned with determination of which Tasks are applicable, or which follow-up Tasks are required, given the state of the Case. Decisions may be triggered by events or new facts that continuously emerge during the course of the Case, such as the receipt of new documents, completion of certain Tasks, or achieving certain Milestones. Individual Tasks that are planned and executed in the context of the Case might be predefined procedural Processes in themselves, but the overall Case cannot be orchestrated by a predefined sequence of Tasks. Representation of the circumstances and the decision factors in a Case model requires references to data about the subject of the Case. Modeling of constraints and guidance on the actions to be taken in a Case requires the specification of rules that reference the data in the CaseFile.... A new Case model may be defined as entirely at the discretion of

human participants initially, but it should be expected to evolve as repeatable patterns and best practices emerge." (OMG, 2013) A CMS supports communication of the system with different business actors. The system reaction to the requests of actors is deterministically defined by business rules. It should not contain implementation details. Business rules are formulated using visible states of the system expressed with data. Therefore, the behaviour modelling method should
(1) result in a deterministic models,
(2) present only visible system states and
(3) enable expressions with relevant data and data transformation.
The distinguished property of these systems is the frequent need of change and subsequent testing due to the changes of law regulations and business optimization. Modelling approaches which facilitate independent modelling of a change and its composition with the old model can reduce modelling and testing effort.

In this paper, we address the problem of model-based testing of CMSs. First, we analyse the modelling techniques in order to select the one that is the most suitable for CMSs. Second, we propose a method of using the chosen modelling technique for test generation and the corresponding algorithms.

2 MODELLING APPROACHES TO MODEL-BASED TESTING

The first research question that we answer in the paper is: Is there a modelling approach most suitable to the domain of Case Management Systems?

We start our analysis with the approaches standardised in the UML (OMG, 2003).

2.1 Modelling Approaches in the UML

Use Cases are used to describe the scenarios of interactions between the system and the environment. *Use Cases* are presented in a combination of the natural language and informal diagrams. The informal and partial use case does not cover system requirements, does not use data, does not support the composition of models.

Interaction Diagrams (Sequence Diagrams and Communication Diagrams) can be used to express interaction of communicating objects during their lifetime. Normally, one such diagram presents one sequence of interaction. If decision points are used, an interaction diagram can present a restricted number of sequences. A set of interaction diagrams usually presents only a subset of possible sequences. Composition of interaction diagrams is not defined in the UML. The suggestions for "mechanical" composition (Roubtsova and Kuiper, 2003; Greenyer et al., 2008; Hanenberg et al., 2007) do not guarantee the specification of complete behaviour.

Activity Diagrams provide a flow based modeling medium, similar to Petri Nets (Petri and Reisig, 2008). They are used to show "the sequence and conditions for coordinating behaviors" rather than behaviours of objects. (OMG, 2003). This means that Activity Diagrams are not suitable for representing the behavior of an object model and, as a consequence, the separate modelling of requirements is not always possible.

State Machines exist in two variants: *Behavioral State Machines (BSM)* and *Protocol State Machines (PSM)* (OMG, 2003).

A *Behavior State Machine* usually presents behavior of one class. "Behavior is modeled as a traversal of a graph of state nodes interconnected by one or more joined transition arcs that are triggered by the dispatching of series of (event) occurrences. During this traversal, the state machine executes a series of activities associated with various elements of the state machine." (OMG, 2003) A transition label $e[g]/a$ includes an event e that triggers the transition, a guard g that restricts the condition of a transition firing and the action a that happens if the transition takes place. The composition semantics for *BSMs* defines an event queue which holds the incoming event instances until they are dispatched, an event dispatcher mechanism that selects event instances from the queue, and an event processor which processes dispatched event instances according to the general semantics of the UML state machines. Event instances being results of some actions within the system or in the environment are conveyed to one or more *BMSs* (OMG, 2003). An event is received by a BMS if it is placed on the event queue of the *BMS*. An event is dispatched when it is removed from the queue and delivered to the state machine for processing. If an event is dispatched, it may result in one or more transitions. If no transition is enabled, then the event is discarded. Finally the event is consumed and it is no longer available for processing. The consumption of events depends on the active state of a state machine. If an event triggers a transition in the current state, it is dispatched and consumed. "A state may specify a set of event types that may be deferred in the state. An event instance that does not trigger any transition in the current state will not be dispatched if its type matches with the type of one of the deferred events. Instead it remains in the event queue while another not deferred message is dispatched instead" (OMG, 2003). In other words,

- if an event recognized by a machine that enables a transition in its current state, then it can be consumed;

- if an event recognized by a machine that does not enable a transition in the current, an this event is included in the list of deferred events for this state, it is kept in a queue for later processing;
- otherwise the event is discarded (McNeile and Roubtsova, 2009).

This means that processing of an event by different machines is asynchronous and the result of processing is non-deterministic. If an event can cause firing of two transitions, which transition can happen first is not defined.

Thus, *BSMs* do provide complete behavior descriptions and can provide model based execution needed for MBT. However, *BSM* do not result in deterministic models and do not meet our requirements for the modelling method for MBT of CMS.

The UML Protocol State Machines (PSMs) are used to express the transitions that a classifier can trigger. A *PSM* is a way to define a life cycle of an object, or an order of the invocation of its operations. PSMs can express usage scenarios of object classes and interfaces. The effect actions of transitions are not specified in a *PSM* transition as the trigger itself is the operation. However, pre- and post- conditions are specified, so that the label of a transition is of the form $[precondition]\ event/\ [postcondition]$. The occurrence of an event that a *PSM* cannot handle is viewed as a precondition violation, but the consequent behavior is left open. “The interpretation of the reception of an event in an unexpected situation (current state, state invariant, and pre-condition) is a *semantic variation point*: the event can be ignored, rejected, or deferred; an exception can be raised; or the application can stop on an error. It corresponds semantically to a pre-condition violation, for which no predefined behavior is defined in UML” (OMG, 2003). Unlike *BSMs*, *PSMs* can (to a limited extent) be composed. “A class may have several protocol state machines. This happens frequently, for example, when a class inherits several parent classes having protocol state machine, when the protocols are orthogonal.” (OMG, 2003). However, the semantics of composition for *PSMs* is undefined.

As a result of the above analysis we conclude that none of the UML behaviour modelling approaches are satisfying enough to meet the requirements to modelling Case Management Systems.

2.2 Outside the UML

The MBT approaches often experiment with labeled transition systems with different semantics. For example, the Labelled Transition Systems (LTSs) used by Kervinen at al. (Kervinen et al., 2006) (s^0, S, E, T) (s^0 is an initial state; S - a set of states; E is a set of events; $T \subseteq S \times E \times S$) is a deterministic model. There is no state in which any output transition share the same action name. The LTSs can be composed using the CSP parallel composition (Hoare, 1985). Also there are rules of matching events that are used for synchronous composition.

LTSs used by Kervinen at al. (Kervinen et al., 2006) meet most of our requirements to the modelling method for CMS. Such LTSs are able to provide the basis of MBT implementation. The shortcoming is their abstraction from data.

The Protocol Modelling technique (McNeile and Simons, 2006) overcomes the above mentioned shortcomings of LTSs. Protocol Modelling uses the CSP parallel composition (Hoare, 1985) extended by A. McNeile to enable the composition of models with data (McNeile and Simons, 2006). It is proven in (McNeile and Roubtsova, 2008) that protocol models possess the property called *observational consistency*. This property means that a protocol machine may be added to and deleted from the model or locally changed and the trace behaviour of other protocol machines is not affected by the behaviour of of the added, deleted or modified protocol machines (McNeile and Roubtsova, 2008). We argue that, potentially, observational consistency of protocol models can reduce the modelling and testing effort when the CMS is changed.

3 PROTOCOL MODELLING FOR MBT

Our second research question is: How can a protocol model be used for test generation?

3.1 What is a Protocol Model?

The building blocks of a Protocol Model are protocol machines and events. They are instances of correspondingly *protocol machine types* and *event types*.

A protocol machine type is an LTS extended to enable modelling with data:

$$PM_i = (s_i^0, S_i, E_i, A_i, CB_i, T_i),\ where$$

- s_i^0 is the initial state;
- S_i is a non-empty finite set of states.
- E_i is a finite set of recognized event types e_i, coming from the environment.

- A_i is a finite set of attributes of different types. The set can be empty.
- $CB_i(PM_1,...,PM_n,E_1,...,E_m) =$ $(PM_1,...,PM_n,E_1,...,E_m)$ is a callback function for updating the values of the attributes, states and events of the protocol machines of the protocol model. $PM_1,...,PM_n$ are the protocol machines of the protocol model. $E_1,...,E_m$ are events of the protocol model. The default callback function (the absence of the callback) does not change the model elements.
- $T_i \subseteq S_i \times E_i \times S_i$ a finite set of transitions: $t = (s_x, e, s_y),\ \ s_x, s_y \in S_i,\ e \in E_i$. The set of transitions can be empty. The states may be updated without callback functions. The values of the attributes, states and events may be updated using the callback function only as a result of a transition, i.e. as a result of an event acceptance.

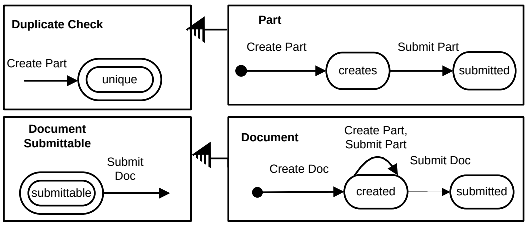
In order to facilitate reuse, protocol machines come in two variants: Objects and Behaviours. Behaviours cannot be instantiated on their own but may extend functionality of objects. In a sense, objects with included behaviours are similar to mixins in programming languages (Bracha and Cook, 1990).

An event type is a tuple

$$e = (EventName, A^e, CB^e)$$

- A^e is a finite set of attributes of the event.
- $CB^e(PM_1,...,PM_n,E_1,...,E_m) =$ $(PM_1,...,PM_n,E_1,...,E_m)$ is a callback function corresponding to this event. A default callback function (the absence of it) does not changes the model elements. The callback function for an event is used if the event calculates attributes of generates other events from the state of the model.

An example of a protocol model is shown in Figure 2. It is a model of a small case management system for handling a document. Unique document parts are submitted to a document. When all parts have been submitted, the document can be submitted. to a document by participants of a project. There are four protocol machines in Figure 2: *OBJECT Part* with included *BEHAVIOUR Duplicate Check* and *OBJECT Document* with included *BEHAVIOUR Document Submittable*. The protocol machines are described textually as sets of their attributes, states and transitions. A behaviour protocol machine is always included into at least one object protocol machine. The *INCLUDES* relation is shown on the top of Figure 2 graphically as an arc with a half-dashed arrowhead. Each of behaviours has the corresponding callback presented as a small Java function. The callback of the behaviour *Duplicate check* returns states *unique* or *duplicate*. The callback of the behaviour *Document Submittable* returns the state *submittable*, *not submittable*. There are four event types in the model presented with their attributes: *Create Part*, *Create Doc*, *Submit Part* and *Submit Doc*.



```
OBJECT Part
NAME Name
INCLUDES   Duplicate Check
ATTRIBUTES Name: String,
STATES created ,submitted
TRANSITIONS @new* Create Part= created,
                  created*Submit Part=submitted

BEHAVIOUR !Duplicate Check
STATES unique, duplicate
TRANSITIONS @any*Create Part=unique

OBJECT Document
NAME Name
INCLUDES DocumentSubmittable
ATTRIBUTES Name:String,
STATES created, submitted, cancelled
TRANSITIONS @new*CreateDoc=created,
              created*CreatePart=created,
              created*SubmitDoc=submitted

BEHAVIOUR !DocumentSubmittable
# Ensures that a document  cannot be
#submitted if it has unfinished Parts
ATTRIBUTES !Document Status: String
STATES submittable, not submittable
TRANSITIONS submittable*SubmitDoc=@any
```

```
EVENT Create Part
ATTRIBUTES
Part: Part,
Name: String,
Document: Document

EVENT Submit Part
ATTRIBUTES
Part :Part,
Document: Document

EVENT CreateDoc
ATTRIBUTES
Document:Document,
Name:String,

EVENT SubmitDoc
ATTRIBUTES
Document: Document,
```

```
public class DuplicateCheck extends Behaviour {
          public String getState() {
                    String myName=getString("Name");
                    Instance[] existingIns = this.selectInState(this.getObjectType(), "@any");
                    for (int i = 0; i < existingIns.length; i++)
               if (existingIns[i].getString("Name").equals(myName) &&
                 !existingIns[i].equals(this))
                    return "duplicate";
                    return "unique";
          }
}
public class DocumentSubmittable extends Behaviour {
  public String getState() {
            boolean allSubmitted =true;
                    Instance[] myParts = this.selectByRef("Part", "Document");
                    if (myParts.length==0)
                    {allSubmitted = false;}
                    for (int i = 0; i < myParts.length; i++) {
                              if ((myParts[i].getState("Part").equals("created")))
                              allSubmitted = false;
                    }
                    return allSubmitted ? "submittable" : "not submittable";
          }
          public String getDocumentStatus() {
                    return this.getState("DocumentSubmittable");
          }
}
```

Figure 2: Protocol Model of a Document that consists of Parts.

3.2 Data Handling and Its Use for MBT

Within the Protocol Modelling, the callback functions are the instruments for data handling. In the ModelScope tool (McNeile and Simons, 2000) supporting execution of protocol models, the callbacks are coded as small Java classes with methods changing and/or returning the values of attributes and states of instances of protocol machines. They may also change attributes of events and generate event instances.

Usually transitions $t \in T_i$ of a protocol machine PM_i enable updates of its own states of the state set S_i. On the other hand, protocol machines can read the states of each other, although cannot change them. This property creates dependencies of protocol machines. The dependency means that one protocol machine needs to read the state of another machine to calculate its own state and/or the attributes. Such calculated states are called *derived states*, which distinguishes them from the ”stored” states denoted in the model (McNeile and Simons, 2006). Callback functions CB_i are used to update attributes and calculate derived states.

There are two possibilities in dependent machines:
(1) The pre-state of a transition can be calculated. The pre-state is similar to a guard calculated in the UML state machines. *BEHAVIOUR Document Submittable* is an example of a pre-state.
(2) The predicted after-state of a transition can be calculated and used to allow or refuse the event. This predicted after-state is a unique feature of Protocol modelling. Both behaviour protocol machines in Figure 2 have the after-states. For example, *BEHAVIOUR Duplicate Check* derives state *unique* or *duplicate* after event *CreatePart* and enables the event only if the *Part* has a *unique* name. If an event is not enabled, the protocol machine is rolled back to the previous state of the model.

As the state of a dependent machine is changed by the corresponding callback, not by a transition, the dependency between protocol machines needs to be taken into account for test generation. Using the information in the callbacks, the derived states of a protocol machine can be visualised in the model with the corresponding stored states with the same names. The state changes, coded in the callbacks, can be presented as transitions. The transformed in such a way protocol machines with derived states, are called connected forms (McNeile and Roubtsova, 2012). They have been used for workflow analysis of protocol models. We propose to use them for test generation. Currently, we perform the model transformation as a manual procedure.

For example, the callback *Duplicate Check* extends behaviour *BEHAVIOUR Duplicate Check.*

The state *duplicate* may appear during the proceeding of event *CreatePart* but it should be rolled back. This means that there is no continuation of the test possible with the duplicated *Part*. In such a situation, the ModelScope tool generates an error message. The working protocol model of *BEHAVIOUR Duplicate Check* contains only one transition (Figure 2).

```
BEHAVIOUR !DuplicateCheck
    STATES unique, duplicate
    TRANSITIONS @any*CreatePart=unique,
```

Using the information from the callback we create the connected form being a non-deterministic protocol machine intended to be used for test generation. It contains two transitions for event *CreatePart* and the third transition for the event *SubmitPart* possible only in state *unique*:

```
BEHAVIOUR !DuplicateCheck
    STATES unique, duplicate
    TRANSITIONS @any*CreatePart=unique,
                @any*CreatePart=duplicate,
                unique*SubmitPart=unique
```

3.3 Analysis of CSP Parallel Composition and It Use for MBT

Another observation concerns the use of the CSP composition.

A protocol model contains only the protocol machine types serving as patterns for creating instances of protocol machines. In the initial state, a protocol model *PM* is a CSP parallel composition of the protocol machines each of which corresponds to a protocol machine type in the initial state *@new*. The instances of protocol machines of each type are created as a result of an acceptance of an event that transits the machine from the state *@new* to some state of the machine life cycle.

In any state, a system model *PM* is a CSP parallel composition of finite set of *instances of protocol machines*.

$$PM = \overset{n}{\underset{i=1}{\|}} PM_i = \quad (s_0, S, E, A, CB, T), \quad n \in N.$$

A Protocol Model *PM* is also a protocol machine, the set of states of which is the Cartesian product of states of all composed protocol machines (McNeile and Simons, 2006):

$s_0 = \bigcup_{i=1}^{n} s_i^0$ is the initial state;

$S = \prod_{i=1}^{n} S_i$ is the set of states;

$E = \bigcup_{i=1}^{n} E_i$ is the set of events;

$A = \bigcup_{i=1}^{n} A_i$ is the set attributes of all machines;

$CB = \bigcup_{i=1}^{n} CB^i$ is the set of callbacks of all machines.

The set of transitions T of the protocol model is defined by the rules of the CSP parallel composition (Hoare, 1985). The rules synchronise transitions T_i of protocol machines. Namely, a Protocol Model handles only one event at a time. An event can be accepted only if all protocol machines having this event in their alphabets are in the state where they can accept this event. Otherwise the event is refused.

As a protocol machine can contain several instances of a certain type, tests should be able to differentiate them. This is possible since each *OBJECT* has a *NAME* attribute by which it can be identified. In our example in Figure 2 all events have only one link to object *Part*. This means that only one instance is used.

Similarly, any *BEHAVIOUR* can be included in several objects. In such cases the corresponding object model contains *INCLUDES* statement (see Figure 2) with the name of included *BEHAVIOUR* type. Although *BEHAVIOUR* does not have a *NAME* attribute, it can be uniquely identified in tests by the *NAME* attribute of the host *OBJECT*.

4 TEST GENERATION

Our proposed Protocol Model Based test generation technique consists of the following steps:

- Creation of object models for all individual protocol machines;
- Iterative generation of the System Protocol Machine, using the CSP composition;
- Generation of test traces for the system protocol machine.

4.1 Creation of Object Models for Individual Protocol Machines

Our test generation starts with building object models containing all such 'first class' model elements as protocol machines (objects and behaviours), events and callbacks as well as their attributes, states, transitions. These models are built by parsing the system protocol model textual representations in Figure 2.

4.2 Iterative Generation of the System Protocol Machine

The fact that the CSP parallel composition of protocol machines is a protocol machine allows us to implement the iterative algorithm for generating the system protocol machine from the models of objects and behaviours obtained at the first step (Algorithm 1).

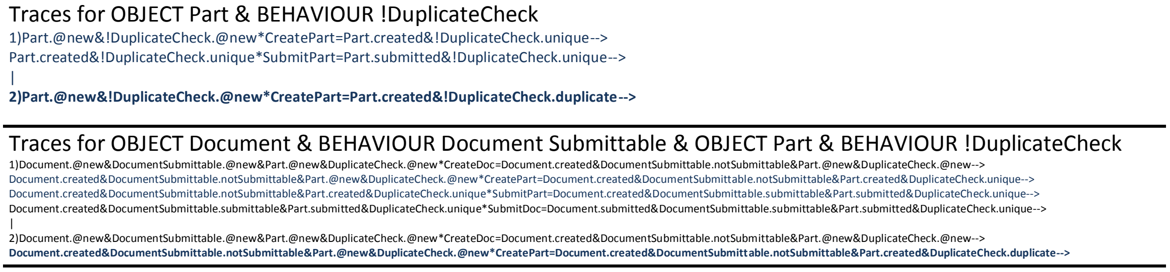
Algorithm 1: Algorithm for CSP parallel composition || of all protocol machines.

```
GIVEN: Set of Protocol machines from the protocol model $\{PM_i\}$, $i = 0, ...n$
RESULT: CSP parallel composition of all protocol machines of the protocol model $SystemPM = ||PM_i \; i = 0, ...n$
Algorithm:
$SystemPM$ = $PM_0$
i = 1
while (i <n) do
    Algorithm 2 for CSP composition || of two Protocol Machines
    $SystemPM$ = $SystemPM||PM_i$
    i++
end while
```

Algorithm 2: Algorithm for CSP composition || of two Protocol Machines.

```
GIVEN:
Object (behaviour) $O_1$ :
Transitions $O_1 : x_i * A = x_j$. Set of events $E_1$. Set of states $S_1$.
Object (behaviour) $O_2$ :
Transitions $O_2 : y_k * A = x_h$. Set of events $E_2$. Set of states $S_2$.
RESULT: CSP parallel composition $O_1||O_2$.
Algorithm:
Find $E_1 \cap E_2$.
for Each $e \in E_1 \cap E_2$ do
    Find transitions of $O_1 : O_1.x_i * e = O_1.x_j$, labeled by $e$.
    Find transitions of $O_2 : O_2.y_k * e = O_2.y_h$, labeled by $e$.
    Construct the transition of the composition of $O_1||O_2$ :
    $O_1.x_i; O_2.y_k * e = O_1.x_j; O_2.y_h$.
    Add the initial and end states of the transition to the set of states of $O_1||O_2$.
    Add $e$ to the set of events of $O_1||O_2$.
end for
for Each $a \notin E_1 \cap E_2$ do
    Find transitions of $O_1 : O_1.x_i * a = O_1.x_j$ labeled by $a$;
    for All states of $O_2 : y$ do
        Construct transition of $O_1||O_2 : O_1.x_i; O_2.y * a = O_1.x_j; O_2.y$;
        Add initial and end states of the transition to the set of states of $O_1||O_2$
        Add $a$ to the set of events of $O_1||O_2$
    end for
    Find transitions of $O_2 : O_2.y_k * a = O_2.x_h$ labeled by $a$;
    for All states of $O_1 : x$ do
        Construct transitions of $O_1||O_2 : O_1.x; O_2.y_k * a = O_1.x; O_2.y_h$;
        Add initial and end states of the transition to set of states of $O_1||O_2$
        Add $a$ to the set of events of $O_1||O_2$
    end for
end for
```

The Algorithm 2 for CSP composition || of two Protocol Machines reflects the synchronization of protocol machines. The rules of the CSP composition say that an event is accepted by the Protocol Model if it is accepted by both protocol machines if they both contain this event in their event sets, otherwise it is refused. This means that if the transitions of two protocol machines are labeled with the same event, then their initial and final states are united to produce a synchronised transition. If an event belongs to the set of only one of protocol machines (in the set of two), then the state of this machine is changed according to



```
Traces for OBJECT Part & BEHAVIOUR !DuplicateCheck
1)Part.@new&!DuplicateCheck.@new*CreatePart=Part.created&!DuplicateCheck.unique-->
Part.created&!DuplicateCheck.unique*SubmitPart=Part.submitted&!DuplicateCheck.unique-->
|
2)Part.@new&!DuplicateCheck.@new*CreatePart=Part.created&!DuplicateCheck.duplicate-->

Traces for OBJECT Document & BEHAVIOUR Document Submittable & OBJECT Part & BEHAVIOUR !DuplicateCheck
1)Document.@new&DocumentSubmittable.@new&Part.@new&DuplicateCheck.@new*CreateDoc=Document.created&DocumentSubmittable.notSubmittable&Part.@new&DuplicateCheck.@new-->
Document.created&DocumentSubmittable.notSubmittable&Part.@new&DuplicateCheck.@new*CreatePart=Document.created&DocumentSubmittable.notSubmittable&Part.created&DuplicateCheck.unique-->
Document.created&DocumentSubmittable.notSubmittable&Part.created&DuplicateCheck.unique*SubmitPart=Document.created&DocumentSubmittable.submittable&Part.submitted&DuplicateCheck.unique-->
Document.created&DocumentSubmittable.submittable&Part.submitted&DuplicateCheck.unique*SubmitDoc=Document.submitted&DocumentSubmittable.submittable&Part.submitted&DuplicateCheck.unique-->
|
2)Document.@new&DocumentSubmittable.@new&Part.@new&DuplicateCheck.@new*CreateDoc=Document.created&DocumentSubmittable.notSubmittable&Part.@new&DuplicateCheck.@new-->
Document.created&DocumentSubmittable.notSubmittable&Part.@new&DuplicateCheck.@new*CreatePart=Document.created&DocumentSubmittable.notSubmittable&Part.created&DuplicateCheck.duplicate-->
```

Figure 3: Traces of the Part with Duplicate check and the model of a Document with one Part.

its transition, and the state of the other machine remains unchanged.

4.3 Generation of Test Traces for the System Protocol Machine

To generate tests we use the set of transitions of *SystemPM* obtained at the previous step. The preconditions are the following:

- Each unique trace has to start from the transition in which the initial state of each object is *@new*. This way a test case can cover the whole life cycle of the objects involved in a trace.

- Each possible transition in a particular trace has to be repeated only once. This way we decrease the number of test steps while still providing complete transition coverage.

Algorithm 3: Generation of test traces for the protocol model.

```
GIVEN: Transitions $T$ of $SystemPM$: $(x,e,y), x,y \in S,\ e \in E$
RESULT: test case traces for $SystemPM$
Algorithm:
Initial state of the system $l$ =”all objects in state @new”.
for each $t \in T$ do
    if The input state of $t$ $x \subseteq l$ then
        Add $t$ to the set of transitions $T_j,\ j = 0,...,k$ possible in
        the initial state ”all objects in state @new”
    end if
end for
for Each $T_j$, j= 0,...,k do
    Apply the Algorithm for Transition Tree Generation for $T_j$
    Make a trace by Traversing Transition Tree for $T_j$
end for
```

Algorithm 4: Transition Tree Generation recursively makes a tree of traces starting with the initial transition T_j.

```
GIVEN:
Transitions $T$ of $SystemPm$
Transition $Tj$ for which precondition 1 holds
Result: test case traces for $T_j$ as an initial transition
Algorithm:
Add $T_j$ to a tree as a node
 remove $T_j$ from $T$ /* to enforce precondition 2*/
for each $t \in T$ do
    if initial state $t$ = end state of $T_j$ then
        Transition Tree Generation for t
    end if
end for
```

Traversing Transition Tree from the initial transition T_j is straight forward. Each trace starts from the initial transition and stops in a leaf for which no transition exists with the initial state equal to the end state in the current leaf. Termination is assured because at each step in the trace the corresponding transition is removed from the finite set T.

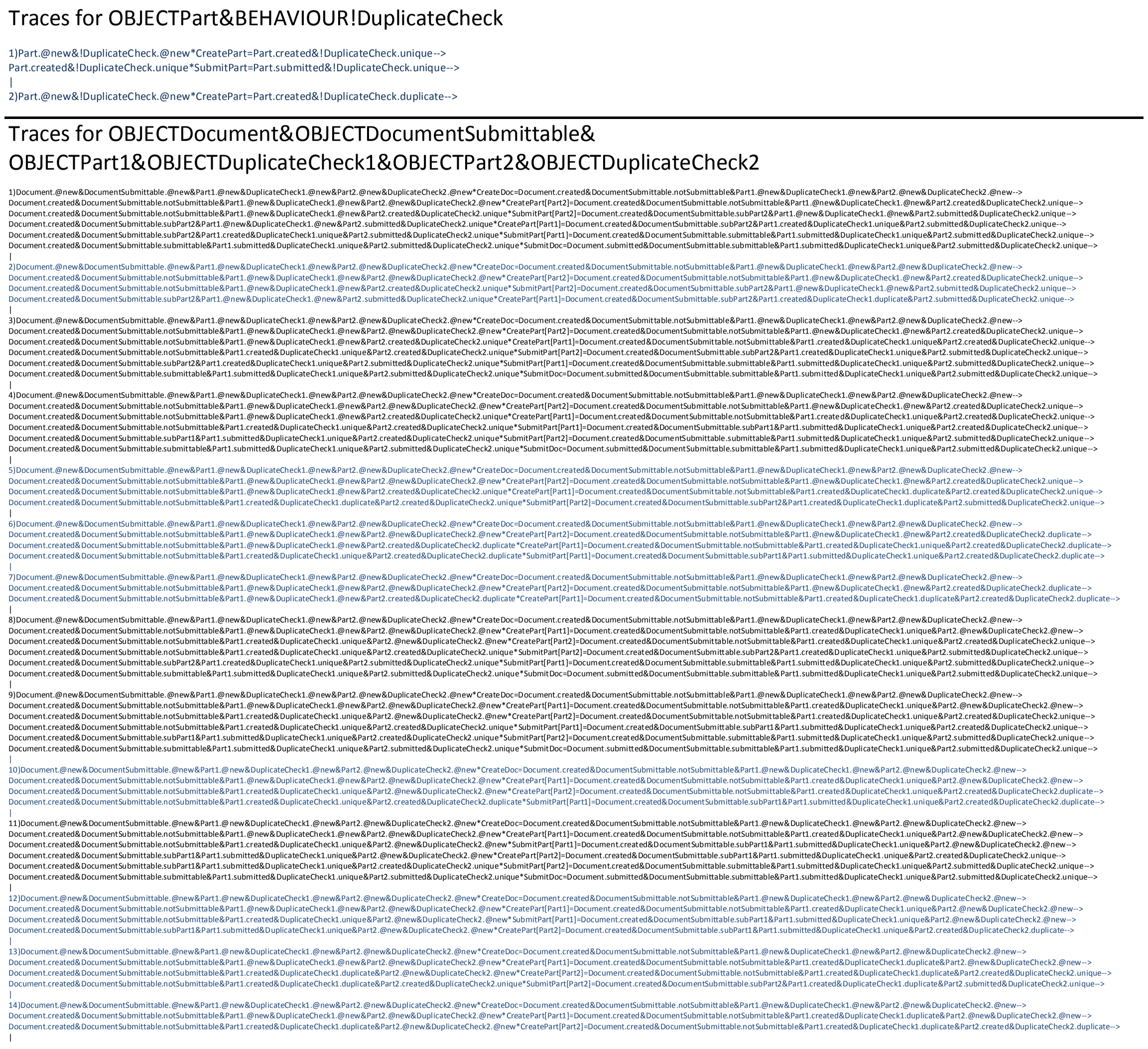
4.4 Results of Test Generation

We implemented the above algorithms as a Java application. As we already mentioned, changeability is an immanent feature of Case Management Systems. In order to support testing of changeability, the traces can be built evolutionary. The task of regression testing (Myers, 2004) is selecting the minimal set of tests required to cover a particular change. Reducing the regression testing can contribute significantly in decreasing the overall testing effort.

Let us assume, that the model for *OBJECT Part* with *BEHAVIOUR Duplicate Check* was created first. After that it was extended to a composed model *OBJECT Document, BEHAVIOUR Document Submittable, OBJECT Part, BEHAVIOUR !DuplicateCheck*.

Traces for *OBJECT Part* with *BEHAVIOUR Duplicate Check* are shown on the top of Figure 3. Traces for the composed model *OBJECT Document, BEHAVIOUR Document Submittable, OBJECT Part, BEHAVIOUR !DuplicateCheck* are show on the bottom of Figure 3.

As we can see, lines *2* and *3* of the trace *1* of the composed model include the trace 1 of the *OBJECT Part* with *BEHAVIOUR Duplicate check*. The states of *OBJECT Document, BEHAVIOUR Document Submittable* are not changed in this trace. Similarly, line 2 of trace 2 includes the trace 2 of *OBJECT Part, BEHAVIOUR Document Submittable*. This means that there is no need to repeat the tests of *OBJECT Part, BEHAVIOUR Duplicate check* for regression testing. The property of observational consistency of Protocol Models works for reduction of regression testing.



```
Traces for OBJECTPart&BEHAVIOUR!DuplicateCheck

1)Part.@new&!DuplicateCheck.@new*CreatePart=Part.created&!DuplicateCheck.unique-->
Part.created&!DuplicateCheck.unique*SubmitPart=Part.submitted&!DuplicateCheck.unique-->
|
2)Part.@new&!DuplicateCheck.@new*CreatePart=Part.created&!DuplicateCheck.duplicate-->

Traces for OBJECTDocument&OBJECTDocumentSubmittable&
OBJECTPart1&OBJECTDuplicateCheck1&OBJECTPart2&OBJECTDuplicateCheck2

1)Document.@new&DocumentSubmittable.@new&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreateDoc=Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new-->
Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreatePart[Part2]=Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.created&DuplicateCheck2.unique-->
Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.created&DuplicateCheck2.unique*SubmitPart[Part2]=Document.created&DocumentSubmittable.subPart2&Part1.@new&DuplicateCheck1.@new&Part2.submitted&DuplicateCheck2.unique-->
Document.created&DocumentSubmittable.subPart2&Part1.@new&DuplicateCheck1.@new&Part2.submitted&DuplicateCheck2.unique*CreatePart[Part1]=Document.created&DocumentSubmittable.subPart2&Part1.created&DuplicateCheck1.unique&Part2.submitted&DuplicateCheck2.unique-->
Document.created&DocumentSubmittable.subPart2&Part1.created&DuplicateCheck1.unique&Part2.submitted&DuplicateCheck2.unique*SubmitPart[Part1]=Document.created&DocumentSubmittable.submittable&Part1.submitted&DuplicateCheck1.unique&Part2.submitted&DuplicateCheck2.unique-->
Document.created&DocumentSubmittable.submittable&Part1.submitted&DuplicateCheck1.unique&Part2.submitted&DuplicateCheck2.unique*SubmitDoc=Document.submitted&DocumentSubmittable.submittable&Part1.submitted&DuplicateCheck1.unique&Part2.submitted&DuplicateCheck2.unique-->
|
2)Document.@new&DocumentSubmittable.@new&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreateDoc=Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new-->
Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreatePart[Part2]=Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.created&DuplicateCheck2.unique-->
Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.created&DuplicateCheck2.unique*SubmitPart[Part2]=Document.created&DocumentSubmittable.subPart2&Part1.@new&DuplicateCheck1.@new&Part2.submitted&DuplicateCheck2.unique-->
Document.created&DocumentSubmittable.subPart2&Part1.@new&DuplicateCheck1.@new&Part2.submitted&DuplicateCheck2.unique*CreatePart[Part1]=Document.created&DocumentSubmittable.subPart2&Part1.created&DuplicateCheck1.duplicate&Part2.submitted&DuplicateCheck2.unique-->
|
3)Document.@new&DocumentSubmittable.@new&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreateDoc=Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new-->
Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreatePart[Part2]=Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.created&DuplicateCheck2.unique-->
Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.created&DuplicateCheck2.unique*CreatePart[Part1]=Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.unique&Part2.created&DuplicateCheck2.unique-->
Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.unique&Part2.created&DuplicateCheck2.unique*SubmitPart[Part2]=Document.created&DocumentSubmittable.subPart2&Part1.created&DuplicateCheck1.unique&Part2.submitted&DuplicateCheck2.unique-->
Document.created&DocumentSubmittable.subPart2&Part1.created&DuplicateCheck1.unique&Part2.submitted&DuplicateCheck2.unique*SubmitPart[Part1]=Document.created&DocumentSubmittable.submittable&Part1.submitted&DuplicateCheck1.unique&Part2.submitted&DuplicateCheck2.unique-->
Document.created&DocumentSubmittable.submittable&Part1.submitted&DuplicateCheck1.unique&Part2.submitted&DuplicateCheck2.unique*SubmitDoc=Document.submitted&DocumentSubmittable.submittable&Part1.submitted&DuplicateCheck1.unique&Part2.submitted&DuplicateCheck2.unique-->
|
4)Document.@new&DocumentSubmittable.@new&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreateDoc=Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new-->
Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreatePart[Part2]=Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.created&DuplicateCheck2.unique-->
Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.created&DuplicateCheck2.unique*CreatePart[Part1]=Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.unique&Part2.created&DuplicateCheck2.unique-->
Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.unique&Part2.created&DuplicateCheck2.unique*SubmitPart[Part1]=Document.created&DocumentSubmittable.subPart1&Part1.submitted&DuplicateCheck1.unique&Part2.created&DuplicateCheck2.unique-->
Document.created&DocumentSubmittable.subPart1&Part1.submitted&DuplicateCheck1.unique&Part2.created&DuplicateCheck2.unique*SubmitPart[Part2]=Document.created&DocumentSubmittable.submittable&Part1.submitted&DuplicateCheck1.unique&Part2.submitted&DuplicateCheck2.unique-->
Document.created&DocumentSubmittable.submittable&Part1.submitted&DuplicateCheck1.unique&Part2.submitted&DuplicateCheck2.unique*SubmitDoc=Document.submitted&DocumentSubmittable.submittable&Part1.submitted&DuplicateCheck1.unique&Part2.submitted&DuplicateCheck2.unique-->
|
5)Document.@new&DocumentSubmittable.@new&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreateDoc=Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new-->
Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreatePart[Part2]=Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.created&DuplicateCheck2.unique-->
Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.created&DuplicateCheck2.unique*CreatePart[Part1]=Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.duplicate&Part2.created&DuplicateCheck2.unique-->
Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.duplicate&Part2.created&DuplicateCheck2.unique*SubmitPart[Part2]=Document.created&DocumentSubmittable.subPart2&Part1.created&DuplicateCheck1.duplicate&Part2.submitted&DuplicateCheck2.unique-->
|
6)Document.@new&DocumentSubmittable.@new&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreateDoc=Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new-->
Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreatePart[Part2]=Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.created&DuplicateCheck2.duplicate-->
Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.created&DuplicateCheck2.duplicate*CreatePart[Part1]=Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.unique&Part2.created&DuplicateCheck2.duplicate-->
Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.unique&Part2.created&DuplicateCheck2.duplicate*SubmitPart[Part1]=Document.created&DocumentSubmittable.subPart1&Part1.submitted&DuplicateCheck1.unique&Part2.created&DuplicateCheck2.duplicate-->
|
7)Document.@new&DocumentSubmittable.@new&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreateDoc=Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new-->
Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreatePart[Part2]=Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.created&DuplicateCheck2.duplicate-->
Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.created&DuplicateCheck2.duplicate*CreatePart[Part1]=Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.duplicate&Part2.created&DuplicateCheck2.duplicate-->
|
8)Document.@new&DocumentSubmittable.@new&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreateDoc=Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new-->
Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreatePart[Part1]=Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.unique&Part2.@new&DuplicateCheck2.@new-->
Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.unique&Part2.@new&DuplicateCheck2.@new*CreatePart[Part2]=Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.unique&Part2.created&DuplicateCheck2.unique-->
Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.unique&Part2.created&DuplicateCheck2.unique*SubmitPart[Part2]=Document.created&DocumentSubmittable.subPart2&Part1.created&DuplicateCheck1.unique&Part2.submitted&DuplicateCheck2.unique-->
Document.created&DocumentSubmittable.subPart2&Part1.created&DuplicateCheck1.unique&Part2.submitted&DuplicateCheck2.unique*SubmitPart[Part1]=Document.created&DocumentSubmittable.submittable&Part1.submitted&DuplicateCheck1.unique&Part2.submitted&DuplicateCheck2.unique-->
Document.created&DocumentSubmittable.submittable&Part1.submitted&DuplicateCheck1.unique&Part2.submitted&DuplicateCheck2.unique*SubmitDoc=Document.submitted&DocumentSubmittable.submittable&Part1.submitted&DuplicateCheck1.unique&Part2.submitted&DuplicateCheck2.unique-->
|
9)Document.@new&DocumentSubmittable.@new&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreateDoc=Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new-->
Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreatePart[Part1]=Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.unique&Part2.@new&DuplicateCheck2.@new-->
Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.unique&Part2.@new&DuplicateCheck2.@new*CreatePart[Part2]=Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.unique&Part2.created&DuplicateCheck2.unique-->
Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.unique&Part2.created&DuplicateCheck2.unique*SubmitPart[Part1]=Document.created&DocumentSubmittable.subPart1&Part1.submitted&DuplicateCheck1.unique&Part2.created&DuplicateCheck2.unique-->
Document.created&DocumentSubmittable.subPart1&Part1.submitted&DuplicateCheck1.unique&Part2.created&DuplicateCheck2.unique*SubmitPart[Part2]=Document.created&DocumentSubmittable.submittable&Part1.submitted&DuplicateCheck1.unique&Part2.submitted&DuplicateCheck2.unique-->
Document.created&DocumentSubmittable.submittable&Part1.submitted&DuplicateCheck1.unique&Part2.submitted&DuplicateCheck2.unique*SubmitDoc=Document.submitted&DocumentSubmittable.submittable&Part1.submitted&DuplicateCheck1.unique&Part2.submitted&DuplicateCheck2.unique-->
|
10)Document.@new&DocumentSubmittable.@new&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreateDoc=Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new-->
Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreatePart[Part1]=Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.unique&Part2.@new&DuplicateCheck2.@new-->
Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.unique&Part2.@new&DuplicateCheck2.@new*CreatePart[Part2]=Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.unique&Part2.created&DuplicateCheck2.duplicate-->
Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.unique&Part2.created&DuplicateCheck2.duplicate*SubmitPart[Part1]=Document.created&DocumentSubmittable.subPart1&Part1.submitted&DuplicateCheck1.unique&Part2.created&DuplicateCheck2.duplicate-->
|
11)Document.@new&DocumentSubmittable.@new&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreateDoc=Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new-->
Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreatePart[Part1]=Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.unique&Part2.@new&DuplicateCheck2.@new-->
Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.unique&Part2.@new&DuplicateCheck2.@new*SubmitPart[Part1]=Document.created&DocumentSubmittable.subPart1&Part1.submitted&DuplicateCheck1.unique&Part2.@new&DuplicateCheck2.@new-->
Document.created&DocumentSubmittable.subPart1&Part1.submitted&DuplicateCheck1.unique&Part2.@new&DuplicateCheck2.@new*CreatePart[Part2]=Document.created&DocumentSubmittable.subPart1&Part1.submitted&DuplicateCheck1.unique&Part2.created&DuplicateCheck2.unique-->
Document.created&DocumentSubmittable.subPart1&Part1.submitted&DuplicateCheck1.unique&Part2.created&DuplicateCheck2.unique*SubmitPart[Part2]=Document.created&DocumentSubmittable.submittable&Part1.submitted&DuplicateCheck1.unique&Part2.submitted&DuplicateCheck2.unique-->
Document.created&DocumentSubmittable.submittable&Part1.submitted&DuplicateCheck1.unique&Part2.submitted&DuplicateCheck2.unique*SubmitDoc=Document.submitted&DocumentSubmittable.submittable&Part1.submitted&DuplicateCheck1.unique&Part2.submitted&DuplicateCheck2.unique-->
|
12)Document.@new&DocumentSubmittable.@new&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreateDoc=Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new-->
Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreatePart[Part1]=Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.unique&Part2.@new&DuplicateCheck2.@new-->
Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.unique&Part2.@new&DuplicateCheck2.@new*SubmitPart[Part1]=Document.created&DocumentSubmittable.subPart1&Part1.submitted&DuplicateCheck1.unique&Part2.@new&DuplicateCheck2.@new-->
Document.created&DocumentSubmittable.subPart1&Part1.submitted&DuplicateCheck1.unique&Part2.@new&DuplicateCheck2.@new*CreatePart[Part2]=Document.created&DocumentSubmittable.subPart1&Part1.submitted&DuplicateCheck1.unique&Part2.created&DuplicateCheck2.duplicate-->
|
13)Document.@new&DocumentSubmittable.@new&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreateDoc=Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new-->
Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreatePart[Part1]=Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.duplicate&Part2.@new&DuplicateCheck2.@new-->
Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.duplicate&Part2.@new&DuplicateCheck2.@new*CreatePart[Part2]=Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.duplicate&Part2.created&DuplicateCheck2.unique-->
Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.duplicate&Part2.created&DuplicateCheck2.unique*SubmitPart[Part2]=Document.created&DocumentSubmittable.subPart2&Part1.created&DuplicateCheck1.duplicate&Part2.submitted&DuplicateCheck2.unique-->
|
14)Document.@new&DocumentSubmittable.@new&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreateDoc=Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new-->
Document.created&DocumentSubmittable.notSubmittable&Part1.@new&DuplicateCheck1.@new&Part2.@new&DuplicateCheck2.@new*CreatePart[Part1]=Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.duplicate&Part2.@new&DuplicateCheck2.@new-->
Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.duplicate&Part2.@new&DuplicateCheck2.@new*CreatePart[Part2]=Document.created&DocumentSubmittable.notSubmittable&Part1.created&DuplicateCheck1.duplicate&Part2.created&DuplicateCheck2.duplicate-->
|
```

Figure 4: Traces generated from the Protocol Model.

For the realistic testing of the model contains *Document* that may be submitted if all its *Parts*, we need at least two *Parts*. The submission of these parts can take place in different order. In the protocol model the number of parts and the submittability is calculated in the callback. Using the information from the callback we build the connected form of the protocol model and generate from it the test cases. The generator produces 14 traces that include the traces of the *Part* with *Duplicate Check*. Figure 4 shows the traces.

The traces 1, 3, 4, 8, 9, 11 (black) include the first trace of the *OBJECT Part* with *BEHAVIOUR Duplicate check* for both *Parts*.

The traces 2, 5, 6, 7, 10, 12, 13, 14 (blue) are terminated by the *BEHAVIOUR Duplicate Check* of one or two *Parts* and do not result in submitted *Document*. The tests of the *OBJECT Part* with *BEHAVIOUR Duplicate Check* are again included into the tests of the new model and should not be repeated. There are lines where the states of other objects and behaviours are not changed. This means that the tests of initial model should not be repeated as regression tests of the extended model.

5 CONCLUSION AND FUTURE WORK

The first research question we addressed in this paper was to find a modelling approach that meets the requirements for model-based testing of highly changeable Case Management Solutions.

As the result of our analysis, we found that the Protocol Modelling technique has the most promising properties as it uses the CSP parallel composition extended with data and possesses the property of observational consistency preserving behaviour of the parts in behaviour of the whole model.

Our second research question addressed the method of using the Protocol Modelling for test generation. We developed a method that combines the technique for manual preparation of a protocol model and the algorithms for automatic test generation from a protocol model. The experiments using our method show that protocol modelling decreases the modelling and testing efforts for regression testing (Myers, 2004) of CMSs. The protocol model presents the external behaviour of a system under the testing. It is different from the system implementation and therefore does not cause error propagation from the system model and code into the tests. Such model changes as behaviour extensions are easily localised as new protocol machines. This reduces the modelling effort for MBT. Thanks to the property of observational consistency a change in the model preserves the tests of the old parts of the system. This reduces the effort for regression testing (Watkins and Simon, 2010). A small illustration of our experiments has been shown in the paper.

The contribution of this paper is shown in Figure 5 as boxes 3 and 4. Our algorithms form parts of the Protocol Model-Based testing tool chain, that we are going to apply for Case Management Systems.

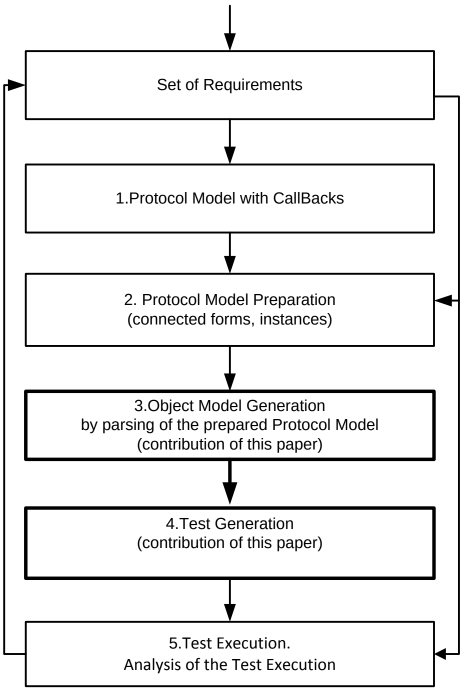

Set of Requirements
1.Protocol Model with CallBacks
2. Protocol Model Preparation
(connected forms, instances)
3.Object Model Generation
by parsing of the prepared Protocol Model
(contribution of this paper)
4.Test Generation
(contribution of this paper)
5.Test Execution.
Analysis of the Test Execution

Figure 5: Protocol Model-Based Testing Methodology.

In future work we are also considering the following research directions:

(1) Partitioning a protocol model into the groups of independent protocol machines with disjoint sets of recognised events. The theory of Protocol Modelling identifies the trace-independent protocol machines and has a proof of observational consistency for them (McNeile and Roubtsova, 2008). Thus, the independent sets of protocol machines should result in independent sets of traces. Each such a group of protocol machines can be used to produce related tests which can be tested separately. This would again facilitate the reduction of testing effort if any of such sets change. In future work we will present our algorithm of partitioning of protocol machines and the methods for selection of a test strategy using partitioning of protocol machines.

(2) Automation of the model preparation and production of test scripts from our test traces. Un-

like other modeling methods, the protocol modeling makes use of callbacks that calculate data and states. Running the callbacks during the process of the generation of tests scripts from the test traces, generation of which is described in this paper, we are going to build executable tests.

REFERENCES

Aalst van der, W. and Weske, M. (2005). Case Handling: a new paradigm for business process support. *Data & Knowledge Engineering*, 52(2):129–162.

Bracha, G. and Cook, W. (1990). Mixin-based inheritance. *OOPSLA/ECOOP '90 Proceedings of the European conference on object-oriented programming on Object-oriented programming systems, languages, and applications*, pages 303–311.

Davenport, T. and Nohria, N. (1994). Case Management and the Integration of Labor. *Sloan Management Review*.

Greenyer, J., Rieke, J., Travkin, O., and Kindler, E. (2008). TGGs for Transforming UML to CSP: Contribution to the ACTIVE 2007 Graph Transformation Tools Contest. University of Paderborn,Technical Report tr-ri-08-287.

Hanenberg, S., Stein, D., and Unland, R. (2007). From aspect-oriented design to aspect-oriented programs: tool-supported translation of JPDDs into code. In *AOSD*, pages 49–62.

Hoare, C. (1985). *Communicating Sequential Processes*. Prentice-Hall International.

Kervinen, A., Maunumaa, M., Pääkkönen, T., and Katara, M. (2006). Model-Based Testing Through a GUI. volume 3997 of *Lecture Notes in Computer Science*, pages 16–31.

McNeile, A. and Roubtsova, E. (2008). CSP parallel composition of aspect models. *AOM'08*, pages 13–18.

McNeile, A. and Roubtsova, E. (2009). Composition Semantics for Executable and Evolvable Behavioural Modeling in MDA. *BM-MDA'09,ACM DL:ISBN: 978-1-60558-503-1*, pages 1–8.

McNeile, A. and Roubtsova, E. (2012). Motivation and guaranteed completion in workflow. *LNBIP*, 142:16–42.

McNeile, A. and Simons, N. (2000). http://www.metamaxim.com/.

McNeile, A. and Simons, N. (2006). Protocol Modelling. A Modelling Approach that Supports Reusable Behavioural Abstractions. *Software and System Modeling*, 5(1):91–107.

Myers, G. (2004). *The Art of Software Testing*. Wiley.

OMG (2003). *Unified Modeling Language: Superstructure version 2.1.1 formal/2007-02-03*.

OMG (2013). Case Management Model and Notation (CMMN). FTF Beta 1. *http://www.omg.org/spec/CMMN/121101*.

Petri, C. and Reisig, W. (2008). *Petri net*, volume 3. Scholarpedia: 3(4):6477.

Roubtsova, E. and Kuiper, R. (2003). Process Semantics for UML Component Specifications to Assess Inheritance. *Electr. Notes Theor. Comput. Sci.*, 72(3):145–159.

UML (2005). Testing profile. *formal/05-07-07,Version 1.0*.

Watkins, J. and Simon, M. (2010). *Testing IT. 2nd ed.* Cambridge University Press.

Model-Driven QoS-aware Approach for the Sensor Network

Assel Akzhalova

Department of Computer Engineering, Kazakh-British Technical University, Tole bi, Almaty, Kazakhstan
a.akzhalova@kbtu.kz

Keywords: Model-Driven, QoS, Optimal, Policy.

Abstract: The key idea of this article is to apply Model-Driven QoS-aware approach to the wireless sensor network that are controlled by the network of ”smart” controllers. The sensors are connected via SPI interface to the controllers installed at the oil wells in order to provide smooth data collection and transmission to meet quality of service requirements. One of the new approaches to implement decentralized adaptation technique, in particular, self-organizing transmission system. The self-organizing system control is entirely dependent on the decision taken at the local level, i.e at the level of system components. However, it is hard to reach global attainability of Quality of Service (QoS) requirements at run-time. In our work, we propose Model-Driven Architecture, the meta-model and its semantics, as the basis for an adaptation framework. The adaptation is realized as an automatic transformation through the policy generation.

1 INTRODUCTION

Service-oriented architecture (SOA) is currently one of the most sophisticated technologies used in modern e-business. The SOA principles illustrate that an integration of heterogeneous business resources such as legacy systems, business partner applications, and department-specific solutions are still emergent topics and there is a need in developing less costly, reusable and interoperable SOA solutions. In practice, there are a lot of problems that make SOA solutions complicated:

Large distributed systems often require huge efforts to deal with legacy in case of incorporating new services into the existing system in order to meet new business requirements. This problem requires research in service composition concerns including developing service description specifications, service discovery, optimal service selection, and binding protocols.

The services might be deployed on different platforms that causes problems in data exchange and increases investments in maintainability of the new system. The solution of this issue suggests elaboration of service functionality implementation providing autonomy and loose-coupling.

Different owners oblige to deal with negotiations and contracts between partners and, therefore, there is a question on how to provide Service Level Agreement (SLA) between participants.

All above are crucial for SOA systems to meet QOS requirements such as security, reliability and performance, especially at runtime.

The one of the ways of resolving the problem is Enterprise Service Bus (ESB) which breaks up the integration logic into easily manageable independent entities. However, the ESBs presented in the market are still have open disputes on the three main topics which are essential requirements stipulated by e-business:

- The SOA systems integration of heterogeneous applications across disparate systems in a flexible and less cost fashion (Martin Keen, 2004).
- Open standards as they are base for successful interoperability across heterogeneous systems (Martin Keen, 2004).
- The SOA systems automatic adaptation in order to meet required constraints and agreements. systems.

An automatic adaptation of SOA can be considered as an automatic service selection. The automatic service selection can be defined by policies that describe contracts between participants containing conditions and actions including penalties in case if the condition will not be met. For example, for the complex service based systems service providers may demand for various charging policies such as payment per resource usage, payment on lifetime services, and also specify available throughput and other constraints. These policies generate Service Level Agreements (SLAs) legally binding contracts that sets constraints on different QoS metrics.

An example of business process that illustrates service selection via policy mechanism is shown in Figure 1. Figure 1 demonstrates run-time system of the oil reservoir automation control framework.

The oil reservoir automation system consists of a number of allocated set of sensors embedded to each oil well equipment that measure various parameters. Each oil well has one intelligent controller for collection and transmission of data measured by sensors. In case of loss of communications with the server, it is necessary to continuously transfer data using an alternative method or, in other words, to guarantee certain level of reliability.

We introduce following assumptions for the node in the sensor network and its attributes:

- Each node has the same logic, transmission and storage of data.
- Each node has the same technical specifications including processor that can perform data (Out) and low level computations and, consequently, received data packed and queued while waiting their processing.
- Each node receives data from several sensors and from neighboring nodes.
- Each node knows how many neighbors (sources) surrounded it and it is able to locally measure the number of packets received from neighbors. Each source has a fair bit of buffer space in a queue. That is each source has own ”channel” not available to other sources. Therefore, this infrastructure is typical example of queuing network.
- The ratio Tp / Ts (passive / sleep timers) determines the cost of energy and the system response to the dynamic changes.

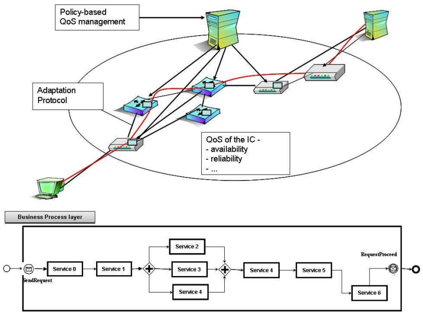


Figure 1: Business process modeling of QoS-aware service system for oil reservoir automation control.

It can be observed especially in the automation of oil and gas industry while monitoring oil fields via controlled sensors and maintaining of high quality of services (QoS) leads to improvement of the oil production rate. In other words, the stipulations of undelayed data collection and transmission pre-determine a choice of the best solutions of optimal production and forecast accuracy.

The control framework can be considered as one that finds optimal set of ICs in order to provide minimum cost and maximum performance for a desirable QoS parameters. An adaptation of the system may be performed through local interactions and, therefore, the overhead is limited by interaction with neighboring nodes. This architecture can be scaled up by allowing the deployment of multiple service instances running on different servers that are devoted for each intelligent controller.

The choice of appropriate services or the process of service selection is defined by policies that show QoS characteristics for the service such as response time, reliability, availability, and throughput. It is essential to design and implement admission control mechanism that will be able to conduct optimal service selection in order to introduce service composition framework supporting QoS. In other words, this framework should represent QoS-aware management and adaptation infrastructure that provides essential service requirements.

Therefore, we suggest that the relationship between possible service composition and QoS constraints will be incorporated into the design of a QoS-aware sensor network architecture: the additional complexity providing significant benefit at runtime through automated policy generation. Section 2 introduce MDE approach for QoS-aware system architecture. Section 3 describes automatic policy generation for the proposed architecture. Section 4 demonstrates Case study. Section 5 compares different techniques that base on policy-aware service composition. Section 6 summarizes contribution and results.

2 MODEL-DRIVEN QoS-AWARE SYSTEM ARCHITECTURE

It is quite often when control of quality of service attributes at run-time is ambitious as there are abundant calculations needed to prepare data for transmission. Moreover, the accuracy of data processing affects the future loads of the distributed system. The main goal of this paper is to apply Model-Driven QoS-aware architecture embedded into the network of controllers which is capable to provide minimum delays while transmitting and processing data and meeting desirable quality of service requirements. QoS-aware management should provide service performance and availability monitoring as well as provision resources based on predefined policies and agreements.

This work presents a model-driven framework for automatic generation of reconfiguration policies. In particular, we suggest that the relationship between possible reconfigurations and QoS constraints should be incorporated into the design of a SOA: the additional complexity providing significant benefit at runtime through automated policy generation. Our focus will be on meeting QoS constraints (performance and reliability) for an overall architecture, what we consider as Service Level Agreements (SLAs).

Figure 2 shows our framework in large. All requests arriving to the system have to be served by all abstract services in finite amount of time according and after serving they leave the system. Every abstract service communicate with a Mediator Service that binds Actual Service from a repository. Actual Services are updating from Service Providers side. The repository is updated by Policy Service which produces optimal selection basing QoS constraints and given SLAs. In fact, Policy Service indicates to each abstract service which Actual Service to pick up.

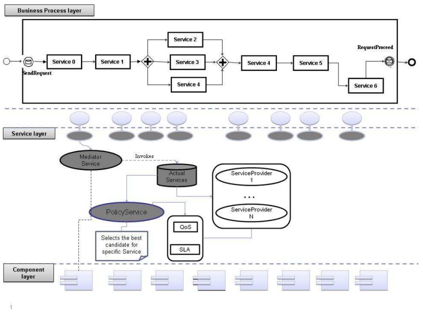


Figure 2: QoS aware service selection and policy generation mechanisms.

In our work we employ a simple generic notion of reconfiguration *policy*, by means of the metamodel extension. The metamodel identifies the architectural roles and relationships that are necessary to construct a model of the monitored system. A distinguishing feature of our metamodel is that it includes

- a QoS constraint language, based on the UML QoS profile, and
- an architectural adaptation policy definition based on dynamic programming approach from optimal control theory that serves both as a specification of how an architecture should evolve in the face of QoS constraint violations for the adaptation engine.

As it can be seen from Figure 2 the dynamic service composition can be implemented by applying policies which are based on QoS requirements and SLAs. The service composition can be considered as model transformation from design-time to run-time abstraction levels.

A policy is modeled at design-time as a possible transformation that an architecture model can undergo, representing possible reconfigurations of service composition. Therefore, service selection allows us to consider the reconfiguration of an architecture as a *transformation* from one SOA model instance to another.

The reconfiguration of an architecture realized as three Model Transformations (MT): MT1, MT2, MT3 (Figure 3).

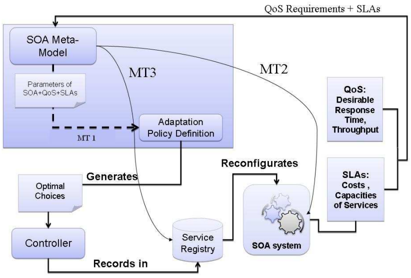


Figure 3: Model transformations for the SOA infrastructure.

The model transformation MT1 is able to automatically frame the problem as a dynamic programming optmization problem, over which our dedicated solver can determine reconfiguration strategies (choices over variant points) as a function of environmental changes. This resulting policy table is then combined with a mapping from choices to actual actions on the implemented system, to provide a runtime adaptation engine.

We employ model transformation MT2 again to extract application metadata from the design time metamodel, with the purpose of understanding how the system is configured at runtime and, consequently, what needs to be monitored:

- Deployment data for individual services (location, interfaces, etc);
- The initial architectural configuration of services (what usage connections exist between services);
- The set of queued interfaces;
- The basic properties that are necessary to compute values of the QoS characteristics used in the model.

A third model transformation MT3 is then used over this monitoring information to change the information associated with individual services in the repository model, for a roundtrip approach to regenerating policies. Policy generation, as outlined in the next section, is expensive and re-generation need not be done frequently as service information does not change often.

3 AUTOMATIC POLICY GENERATION

3.1 QoS-aware Policy Generation

The metamodel is equipped with QoS characteristics and computes the overall cost as a function of time of architectural configurations that has to be minimized. we exhibit two examples of cost definition. As an example of the cost function of the system g can be taken sum of the cost of holding requests in queue at time step t which is formulated by the following expression:

$$g(t) = \sum_{i=0}^{n} c_i^j(t) \cdot q_i(t) \tag{1}$$

where q_i is a queue length of i-th node, c_i^j is a cost of establishing connection between nodes i and j at time Another example is when the cost of the system reflects a negotiation between cost of the system and response time. In this case a sample of the cost function can have the following form:

$$f(g(t), RT) = g(t) + W \cdot max(0, RT_{des} - RT) \tag{2}$$

where RT_{des} desirable response time, RT - response time and W - a positive number $0 \leq W \leq 1$ which is a parameter that defines to which participant of the tradeoff to give more weight. If $W \to \infty$ then second multiplier vanishes. Therefore, the system will have to be balanced between desirable response time and cost of the system.

We consider QoS constraints as a combination of one or more requirements (predicates) P_1, P_2, P_3, P_4 where P_i is one from the set: Reliability, Availability, Throughput, Response Time. We introduce Constraints() method as a rule for terms P_1, P_2, P_3, P_4:

Let
$S :\ System.ActualCharacteristic.QoSCharactersitic$,
$C :\ QoSConstraint.DesirableCharacteristic$,
$\nabla_2 = \{<, \leq, >, \geq\}$

$$P_1(S, \nabla_2, C) = S.SystemResponseTime \to \\ \to Calculate()\ \nabla_2 C.SystemResponseTime.RT \tag{3}$$

$$P_2(S, \nabla_2, C) = S.SystemReliability \to \\ \to Calculate()\ \nabla_2 C.SystemReliability.REL \tag{4}$$

$$P_3(S, \nabla_2, C) = S.SystemAvailability \to \\ \to Calculate()\ \nabla_2 C.SystemAvailability.AVL \tag{5}$$

$$P_4(S, \nabla_2, C) = S.SystemThroughput \to \\ \to Calculate()\ \nabla_2 C.SystemThroughput.Thrpt \tag{6}$$

where Calculate() function computes response time, reliability, availability and throughput.

Therefore, we can expose QoSConstraint.Constraints() as any combination L composing of following pair of predicates defined in BNF form:

$$L = P_a(S, \nabla_2, C) | L \nabla_1 L \tag{7}$$

where $\nabla_1 \in \{\vee, \wedge\}$, and $a \in \{1, 2, 3, 4\}$.

Basing on QoS requirements, an adaption of the system happens to adjust the system to the appropriate performance objective. For instance, the system adjustment objective can be formulated as a rule: ”The system has to have an availability of 99.9% during the business hours of weekdays”. This rule can be expressed according to (5):

$$AVL = 0.999$$

Another example of the rule to conform the system by reliability and response time requirements may sound as: ”The system must be reliable no less than 95% and throughput of the system has to be no less 700 messages per second”.

$$(REL \geq 0.95) \wedge (Thrpt > 700)$$

Therefore, our for given Constraints() we have to find the best set of connected nodes for each node while keeping minimum CostFunction() of the system.

In (Akzhalova and Poernomo, 2010) we proposed MDA for SOA architecture which uses automatic policy generation implemented as optimal service selection framework. We employ similar approach for the sensor network QoS-aware framework.

After cost model has chosen and QoS requirements are determined the model transformation MT1 automatically changes the system to adjust the system to desirable performance level. The system adaptation happens by calling Reconfigure() selftransformation to make the system satisfy to desirable QoS characteristics which are pre-defined in QoSConstraints. Reconfigure() generates Policy which is used then for a Binding appropriate Service to the node. In fact, Reconfigure() produces Policy as a product of the following transformation:

$$\textbf{Reconfigure:} System \times QoSConstraints \to Policy \tag{8}$$

where reconfiguration of System is evaluated by its cost model defined by CostFunction.

Every TimeStep when System violates QoSConstraints, Recongigure() defines Service.ID that has to be bound for each node. We designate a candidate Service as $\{Policy(TimeStep) = ID,\ ID = 1,...,NumberofServices\}$.

To find best candidate service at each time step:

$BestPolicy(TimeStep) \in \{Policy(TimeStep) = ID,$
$ID = 1,...,NumberofServices\}$
that satisfies to QoS constraints:

$$Constraints() \equiv true$$

and gives a minimum to an overall cost of the System:

$$System.CostFunction(Policy) \rightarrow \min, \tag{9}$$

where System changes its reconfiguration according to System.SystemConstraints():

$$SystemConstraints(TimeStep, Policy(TimeStep)) \tag{10}$$

The problem of optimal adaptation at time t is one of choosing the best server from the directory for each i-th service. That is, it is one of finding the best function set $\mathrm{u}_i(t)$ that provides the lowest overall cost while meeting desirable response time. Therefore, in this case the formulation of optimal control problem will be derived from :

To find an optimal control:

$\overline{\mathrm{u}}^*(t) \in U^*(t) = \{\mathrm{u}_i(t) = j : j = 1,...,m, RT(\overline{x}) < RT_{des}\}$
that gives a minimum to functional:

$$J(\overline{x},\overline{\mathrm{u}}) \rightarrow \min, \tag{11}$$

where the configuration of the system is defined by system transformation:

$$\begin{aligned} &\overline{x}(t+1) = F(\overline{x}(t),\overline{\mathrm{u}}(t)),\\ &\overline{x}(t) \in D,\\ &t = 0,...,T-1 \end{aligned} \tag{12}$$

The problem (sec5:eq11) - (trans1) is constrained nonlinear optimal control problem. There is no exact analytical decision of the formulated problem. In order to solve the problem it is necessary to use some of numerical optimization approaches. In the next subsection we give basics of iterative numerical methods to solve the problem and convergency definition.

There is no still agreement on the best approach of solving of the problem 12, however, traditionally, among popular methods of solving such problem are dynamic programming algorithm, genetic algorithm, simulated annealing and others. We apply dynamic programming algorithm, an iterative approach, to solve the formulated problem.

3.2 Best Candidates Selection Algorithm

Dynamic programming is a general approach to solve optimization problems that involve making a sequence of interrelated decisions in an optimum way. First, the problem is divided into subproblems (stages) where each stage has a number of states. The control (decision) at each stage updates the state into the state for the next stage. The main idea of the approach is that at given current state the optimal decision for the remaining stages is independent of decisions made in previous states.

In particular, the algorithm minimizes the sum of the cost incurred at the current stage and the least total cost that can be incurred from all subsequent stages, consequent on this decision. This principle is known as the Bellman's principle of optimality (Bellman, 1957) and dynamic programming algorithm consists of the following steps:

Dividing into Stages. The stages here related to time (hence the name is dynamic programming) and they are solving backward in time. In other words, we consider T stages: $k = T, T-1, ..., 0$.

Defining States at Stage. Each stage has a number of states each of which indicates a candidate server.

The value set of state variable x^k at stage k is the state set at stage k: $x^k = \{x(k),...,x(T)\}$ which is a solution of the problem:

$$\begin{aligned} &x^k(t+1) = F(x^k(t),u^k(t)),\ \ t = k,...,T-1\\ &x^k \in D^k = \{\overline{x}(t) \in D : t = k,...,T\} \end{aligned} \tag{13}$$

and correspondent control at stage k is defined as following:

$$\begin{aligned} &u^k = \overline{u}, u^k \in U^K,\\ &U^k = \{\overline{u}(t) \in U :\ \mathrm{t = k,...,T}\}. \end{aligned} \tag{14}$$

Decision at Stage. The decision at a stage updates the state at a stage into the state for the next stage. Therefore, we calculate cost at current stage k:

$$\begin{aligned} &g^k(x^k(t),\overline{\mathrm{u}}(t)),\\ &u^k \in U^K,\ \ x^k(t) \in D^k\\ &t = k,...,T \end{aligned} \tag{15}$$

and we find the minimum of cost functional:

$$\begin{aligned} &J^k(\overline{x}^k,\overline{\mathrm{u}}^k) = \sum_{t=k}^{T} g(\overline{x}(t),\overline{\mathrm{u}}(t)),\\ &B^k(x^k,u) = \min_{u \in U^k} J^k(x^k,u),\\ &x^k \in D^k. \end{aligned} \tag{16}$$

where $k = T, T-1, T-2, ..., 0.$. The function B^k refers to Bellman's function.

Recursive Value Relationship. According to the fundamental dynamic programming principle of optimality given the current state, the optimal decision for the remaining stages is independent of decisions made in previous states. In other words, the optimum decision at stage k uses the previous found optima. That is recursive relationships means that a cost functional appears in both sides of the following equation:

$$B^k(x^k,u) = \min_{u \in U^k} \{g^k(x^k,u) + B^{k+1}(x^k,u)\}$$
$$u \in U^k, \;\; k = T, T-1, ..., 0. \qquad (17)$$

In the meantime, we find optimal control u^k, $k = T, ..., 0$.

Therefore, following this algorithm we find optimal policies that minimizes cost function for a given constraint.

4 EXPERIMENTS: DEPENDENCY BETWEEN SERVICE CAPACITY AND PERFORMANCE

The presented automatic policy generation approach to select best candidates for the SOA system. It was implemented as a middleware using Java JDK1.6, XML and Eclipse Modeling Tools (Eclipse Classic 3.5.2) on Intel(R)Core(TM)2Quad CPU/2.66Ghz/2.66Ghz/RAM3.23GB. A general scheme of the project can be represented in the following class diagram (Figure 4).

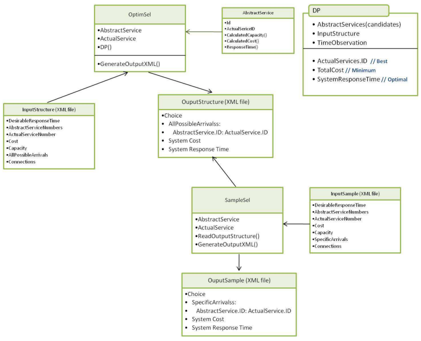


Figure 4: Class diagram of the optimal service selection implementation.

OptimSel class is responsible for generation of optimal choices for given input parameters of SOA system. Input parameters of SOA system are collected by InputStructure Class which is XML file containing QoS requirements such as Desirable Response Time, data about Actual Services number and their Capacities and Costs, Connections between services and all possible Arrivals. Those inputs are used then for a series of experiments in order to test different case studies that will be presented in the next sections. OptimSel reads (parses) InputStructure file and produce OutputStructure XML file by means of DP (dynamic programming) Method.

DP is the Method that performs dynamic progarmming algorithm using Observation Time as a number of states that was described in previous Chapter. DP returns Best Actual Service for each Abstract Service that bring to the system minimum Total Cost and Response Time of the system that is not violating Desirable Response Time.

After calling and executing DP OptimSel produces OutputStructure by calling GenerateOutputStructure() method. GenerateOutputStructure() creates XML file by parsing results to specific attributes. In fact, OutputStructure file contains optimal choices for each Service per each Arrival in order to proceed efficiently arriving to the system requests.

The diagram in Figure 4 contains SampleSel, a derived class from OptimSel, that is used for each case study. For instance, if we simulate the SOA system for different distribution of arrivals we use policies generated and stored in OutputStructure XML and generate table specifically for those arrivals. The table for those arrivals is OutputSampleXML file.

In order to test the system for varying values of Desirable Response time, Capacities, Costs or Connections it has to be generated relative OutputStructure XML files for each case.

The purpose of this experiment is to find out how capacities of the service affect to the cost and performance of the system.

The system is composed of $n = 8$ semantic services and each service has $m = 6$ independent services. Assume that the service at each Semantic service has cost defined by Cost c_i^j and Capacity $\tilde{\mu}_i^j$ (i = 1, .., 8; j = 1, .., 6). The values of Cost and Capacities are shown in Table 1 and Table 2, respectively.

In order to conduct the test we build new tables of capacities which range over average absolute deviation *MD* from previous capacities. An average absolute deviation is also often called Mean absolute Deviation (MD) is the mean of the absolute deviations of a set of data about the data's mean. The *MD* of the set data size *n* is defined by:

Table 1: Maximum capacity of Services at each Semantic Service (requests per second).

ID of the Service	**ID of the Semantic Service**							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	50	55	55	57	65	57	65	67
2	51	60	65	58	67	60	67	68
3	56	67	71	64	70	63	72	70
4	60	70	72	68	72	65	74	71
5	80	85	90	70	80	68	75	75
6	85	90	120	100	120	100	85	84

Table 2: The cost of the Service for each Semantic Service (unit of money per request).

ID of the Service	**ID of the Semantic Service**							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	10	11	15	10	15	16	12	14
2	20	21	23	20	23	25	18	15
3	30	31	35	28	35	30	23	20
4	40	42	45	35	42	40	35	30
5	50	52	55	43	44	41	40	40
6	52	55	60	45	55	45	51	50

$$MD = \frac{1}{n}\sum_{i=1}^{n}\{\alpha_i - \overline{\alpha}\}$$

where $\overline{\alpha}$ - the mean of the distribution:

$$\overline{\alpha} = \frac{1}{n}\sum_{i=1}^{n}\alpha_i$$

Basing on investigation of relationship between *MD* variation and performance of the system we may take this dependency as the next step to study cost functional of the system that can adjust model validation. We consider a system that process Poisson distributed requests (Figure 5) that has to adapt itself to meet $RT < RT_{des}$ constraint where RT_{des}=4 seconds.

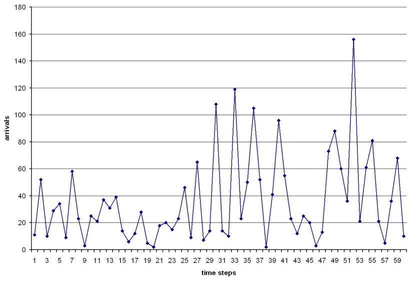


Figure 5: Arrivals to the system.

In fact, handling different capacities of the service, we obtain three systems with services that have different Capacities and Costs. In other words, one can represent those variations as different offers from a number of service providers. Therefore, we examine how different offers may affect to the overall responsiveness and expenses of exploiting systems. In particular, capacities of systems where $MD = -11$, $MD = +11$ are defined in Table 3 - Table 4. We leave the cost of the services fixed (see Table 2).

According to given data we apply dynamic programming algorithm that finds optimal policies for each system. Analyzing how capacities will affect to response time of the system and total cost on Figure 6 and Figure7 we may discover that in general evolution of the systems with different capacities have similar dynamics but different amplitude. For instance, response time obtained by dynamic programming algorithm with higher capacities $+MD$ has the smallest values in comparison with one that was produced by applying services with lower capacities.

Instead the policy that had to select among services with capacities $-MD$ from original transforms the system to the state when one has slowest responsiveness. In other words, the new configuration has violated desirable response time. Presumably, existing resources were not sufficient to tackle with 156 requests at time step 52. At the same time, this outcome shows that in order to apply dynamic programming there is needs to put a constraint on capacities distribution in advance to avoid non-optimal solutions.

Surprisingly, the maximum response time returned by the policy with originally capacities is very close to the value of response time of the system with capacities $+MD$. For instance, in case of highest workload at time step 52 when 156 requests had to be proceed. In other words, having same cost of the service and capacities that differ on 11% from *MD*, we observe that dynamic programming produces re-

Table 3: Maximum capacity $+MD$ of Services at each Semantic Service (requests per second).

ID of the Service	ID of the Semantic Service							
	1	2	3	4	5	6	7	8
1	61	66	66	68	76	68	76	78
2	62	71	76	69	78	71	78	79
3	67	78	82	75	81	74	83	81
4	71	81	83	79	83	76	85	82
5	91	96	101	81	91	79	86	86
6	96	101	131	111	131	111	96	95

sponse time as close as possible to the desirable one. Therefore, the constraint formulated as inequality for dynamic programming is the objective to reach as near as possible. However, the picture of costs of cases with original and capacities $+MD$ is dissimilar.

In particular, as it can be observed in Figure7 the cost of the system in case of exploiting services with higher capacities, was the lowest one. In contrast to the similarity of response times of the system with original capacities and capacities $+MD$, the cost of the system with original capacities at time step 52 is the highest one. In the meantime, when system handles services with lowest capacities $-MD$ the cost of the system was expensive as it had to afford a range of workloads by using less efficient resources.

Table 5 displays maximum achieved response time for different systems and their total cost. As it can be studied from the Table and examining all accommodated systems we have detected that the best variant in terms of minimum cost and response time was generated by policy that operates with services with highest capacities.

As it was mentioned at the beginning of this test, Table 2 remains having constant values of service cost for all considered systems. We may make preliminary assumption that total cost of the system decreases while handling services when the value of their capacities increases. Therefore, there is an inverse relationship between performance of existing resources (services) and cost and response time of the system.

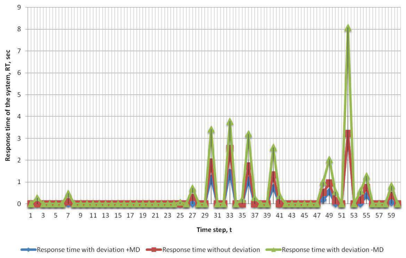


Figure 6: Dynamics of response time for different sets of capacities.

Summarizing the case study, we have investigated an influence of changing capacities to the system characteristics:

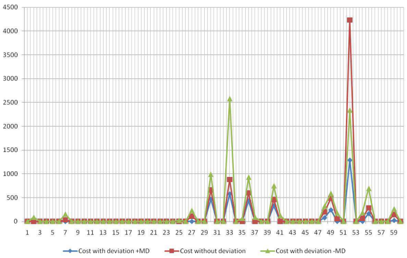


Figure 7: Dynamics of cost of the system for different sets of capacities.

- The experiments showed a considerable impact of changing capacities to the cost of the system. and response time at given constraints.
- If the deviation MD is bigger than 20% then the response time of the system differs significantly than one produced handling original service capacities.
- Employing dynamic programming to the systems with capacities that have $MD + 11\%$, the algorithm gives maximum response time that is very close to one obtained by applying original capacities. In fact, in both cases it attempts to approach to the desirable response time.
- On other hand, if value of capacities less than enough then constraint might be violated.

These outcomes can deliver essential recommendations when system's resources struggle with workloads and their utilization changes from current level. At the same time, these results can used by service providers to install required service contract with service consumers.

5 DISCUSSION

Lymberopoulos Leonidas, Emil Lupu and Morris Sloman in (Lymberopoulos et al., 2003) propose a framework that supports automated policy deployment and

Table 4: Maximum capacity $-MD$ of Services at each Semantic Service (requests per second).

ID of the Service	ID of the Semantic Service							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	39	44	44	46	54	46	54	56
2	40	49	54	47	56	49	56	57
3	45	56	60	53	59	52	61	59
4	49	59	61	57	61	54	63	60
5	69	74	79	59	69	57	64	64
6	74	79	109	89	109	89	74	73

Table 5: The maximum response time and cost of the system for different types of policies.

Type of system	Maximum response time, Max(RT)	Total cost of the system
Services with capacities $-MD$	8.069	10591
Services with original capacities	3.232	8209
Services with capacities $+MD$	3.225	3616
Services with capacities $+2*MD$	2.1099	2494
Services with capacities $+3*MD$	1.4314	1607

flexible event triggers to permit dynamic policy configuration. Basically, the authors developed PONDER policy language for adaptation at the service layer to select and modify policies at the network layer (Damianou et al., 2001). The language provides reuse by supporting definition of policy types, which can be instantiated for each specific environment. The proposed adaptation is run-time and it also allows to build a new adaptation strategies that can be incorporated into the management system by adding new policies which react to different events using the existing policy actions or by replacing existing policies with new versions, which either implement new actions on the managed objects or new actions on the Policy Management Agents. In general, an adaptation is provided in one of the following ways:

- by dynamically changing the parameters of a QoS policy to specify new attribute values for the run-time configuration of managed objects;
- by selecting and enabling/disabling a policy from a set of pre-defined QoS policies at run-time.

The advantage of the presented approach is that parameters of the selected network QoS policy are calculated and set at run-time. The authors addressed the future research on developing techniques that provide policy specification and adaptation across different abstraction layers; and to develop tools and services for the engineering of policy-driven systems.

The research of QoS policy based Web Service selection conducted by (Wang et al., 2006) attempts to generalize QoS contract specification, establishment, and monitoring for Service Level Management into one large framework. The authors introduce a QoS management architecture consisting of component services, their interactions, and interfaces with external services such as real-time host and network condition monitoring (through COTS Monitoring tools like Empirix OneSight/FarSignt) (Wang et al., 2005). The framework includes a tool for end users to generate, modify and validate QoS specifications in the given language. The tool facilitates generation of the QoS specifications without requiring the user to remember the supported set of QoS characteristics and their value domains. The Monitoring Service registers condition predicates with the Diagnostic Service, which returns with notifications after that Monitoring Service updates the corresponding data in Maintenance Service, which in turn activates some Adaptation Mechanisms defined in the policy.

Therefore, the framework provides reusable concepts and processes to facilitate QoS contract establishment and monitoring through contract negotiation, resource management, diagnostics and adaptation mechanisms. However, as the authors note in (Wang et al., 2006), it has to be done the further research and development of dynamic QoS-driven resource management algorithms for Service Level Management.

This work presents a way to tackle with above problems by introducing model-driven approach to SOA together with the optimal control technique as transformation for the SOA meta-model in order to automatically reconfigure the system in less costly way.

6 CONCLUSIONS

The policy generation was formulated as an optimal control problem which allows automatically generate appropriate configuration of the service-oriented system to meet QoS constraints. It means that proposed approach is generic in terms of choosing different level of QoS requirements and applying different techniques to solve the optimal control problem. The model is extensible as the QoS requirements may be included during system design. We have offered dynamic programming approach as the solution of the formulated problems. The case study investigates how different parameters of the service-oriented system and constraints affect to the performance and dynamics of system utilization. We have evaluated obtained results and formulated recommendations and best strategies for employing dynamic programming approach to dynamically adapt SOA according to desirable QoS characteristics.

REFERENCES

Akzhalova, A. and Poernomo, I. (2010). Model driven approach for dynamic service composition based on qos constraints. *Services, IEEE Congress on*, 0:590–597.

Bellman, R. (1957). *Dynamic Programming*. Princeton University Press.

Damianou, N., Dulay, N., Lupu, E., and Sloman, M. (2001). The ponder policy specification language. In Sloman, M., Lobo, J., and Lupu, E., editors, *POLICY*, volume 1995 of *Lecture Notes in Computer Science*, pages 18–38. Springer.

Lymberopoulos, L., Lupu, E., and Sloman, M. (2003). An adaptive policy-based framework for network services management. *J. Netw. Syst. Manage.*, 11(3):277–303.

Martin Keen, Amit Acharya, e. a. (2004). *Patterns: Implementing an SOA Using an Enterprise Service Bus*. IBM Corp., Riverton, NJ, USA.

Wang, C., Wang, G., Wang, H., Chen, A., and Santiago, R. (2006). Quality of service (qos) contract specification, establishment, and monitoring for service level management. In *EDOCW '06: Proceedings of the 10th IEEE on International Enterprise Distributed Object Computing Conference Workshops*, page 49. IEEE Computer Society.

Wang, G., Wang, C., Chen, A., Wang, H., Fung, C., Uczekaj, S., Chen, Y.-L., Guthmiller, W. G., and Lee, J. (2005). Service level management using qos monitoring, diagnostics, and adaptation for networked enterprise systems. In *EDOC '05: Proceedings of the Ninth IEEE International EDOC Enterprise Computing Conference*, pages 239–250. IEEE Computer Society.

SHORT PAPERS

Modelling Information Systems Using Nomis
A Practical View of Its Aplication and Its Insights to Business Processes

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.

Abstract: NOMIS – **NO**rmative **M**odelling of **I**nformation **S**ystems presents a new way of think, model and develop information systems. This new approach has its foundation in a simple and specific ontology known as Human Relativism (HR). HR philosophical stance acknowledges the *human element* central role within any information system (IS) leading us to use *observable human actions* as the IS kernel modelling element. Using *observable human actions* is claimed as a step towards achieving the desired modelling objectivity and precision that most natural sciences and engineering have. NOMIS has also its roots on three IS socio-technical approaches, namely Organisational Semiotics, the Theory of Organised Activity and Enterprise Ontology from where its specific vision and views of IS are *inspired*. Modelling IS with NOMIS can be done by representing NOMIS Vision with the NOMIS Models - a set of diagrams and tables using the NOMIS provided notation or, otherwise, a set of UML profiles created for it. In this paper we provide an overview of NOMIS and some modelling application examples that intend to highlight some new and important concepts introduced by NOMIS. Our focus will be in the most innovative aspects and their relevance to business systems understanding and modelling.

1 INTRODUCTION

Information systems failure was a popular research topic two decades ago (e.g. Sauer, 1997). Nowadays, computerized systems (CS) have evolved, people got familiarized with them and, many issues now go unnoticed. Some of them are revealed when we want do something with a CS and we find out that that particular action is not available, or when we don't know how to do it because there is no information to help us and many other common situations. Also Information System Development (ISD) and, in particular, computer programming is still lacking a solid theoretical foundation. Requirements are gathered with no solid rules; programs are developed based mostly on practice guided by experience and established software patterns, programming structures and data models have many different flavours depending on the implementation.

NOMIS – **NO**rmative **M**odelling of **I**nformation **S**ystems presents a new way of think, model and develop information systems (IS) that intends to improve modelling objectivity and precision. This is accomplished by: (1) adopting a new ontology named *Human Relativism* that recognizes the central role of the human element within an IS, the unpredictability factor it introduces, and a way of reduce this unpredictability; (2) proposing a new vision of IS composed by different views *inspired* by the ideas of three known socio-technical approaches namely Organisational Semiotics (Liu, 2000), the Theory of Organized Activity (Holt, 1997) and Enterprise Ontology (Dietz, 2006); (3) defining a modelling notation and a set of diagrams to represent NOMIS Vision and views.

In this paper NOMIS Foundations, including HR, NOMIS Vision and NOMIS Models, will be briefly presented. The focus and contribution, however, will be in the most innovative aspects of NOMIS approach and its effects on modelling of business systems. This will be shown using a simple case study that will be modelled using NOMIS Models where some key concepts will be highlighted. The examples will highlight some issues of current business modelling and some neglected and new aspects of it.

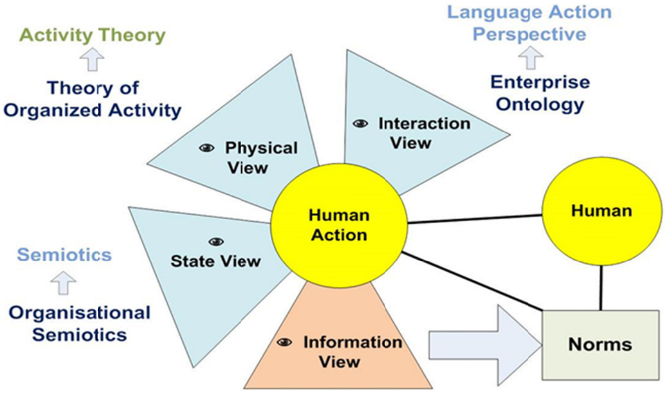


Figure 1: NOMIS Vision – its views and foundational theories.

2 NOMIS FOUNDATIONS

2.1 Human Relativism

Human Relativism (HR) (Cordeiro et al, 2009) is a philosophical stance that differs from *objectivism* by recognizing a reality dependent on the observer. This *experienced reality* in HR does not deny the existence of a single objective reality, therefore does not adhere to *subjectivism* as well. Dependency on the observer or human subject is of utmost importance for information systems (IS) where the first central piece is the *human element.* Realizing that each person has a different view of the IS which is grounded in her experience, emotions, perception, understanding, values, knowledge, etc. is a first step to acquire the necessary precision needed to accurately understand, model, design and develop IS solutions. A second central piece of IS is information, a misunderstood concept (see, for example, Falkenberget al, 1996) that is also dependent on the human element.

From this view computers and computer systems play a secondary supportive role. The *essential* IS is a *human system*.

HR introduces the important notion of *observability* seeking to deal with the unpredictability originated from a reality dependent on the subject. From a human perspective, information is obtained by an individual after a first step of *perception*, where reality is aquired through the human sensitive system and a second step of *interpretation, an inter-subjective* process of meaning making. According to HR, the result of the first step determines the *observable* reality, that should be free from interpretation and can be made objective. Therefore, HR makes the following key assumption:

Anything that is observable will be more consensual, precise and, therefore more appropriate to be used by scientific methods.

Within IS, the human element is the source of unpredictability that constraints the universality of concepts and the correct and adequate gathering of actions and processes. In fact, in IS, information depends on individual interpretation, also performed business actions depend on human performers.

The corollary obtained from HR is that the focus on *observable elements* will reduce the ambiguity and unpredictability of IS and IS modelling in particular. As the human element is the source of unpredictability, we should focus in its observable aspects namely *observable human actions*. These exclude inter-subjective actions, like intentions, judgements, values or decisions, at least their hidden and subjective parts. Also, dealing with information, attention and distinction should be made between the observable part, the physical things that carry the information and, the mental aspects of its acquisition.

2.2 NOMIS Vision

NOMIS theoretical foundations came from three different IS socio-technical approaches, namely the Theory of Organized Activity (TOA) (Holt, 1997), Enterprise Ontology (EO) (Dietz, 2006) and Organisational Semiotics (OS) (Liu, 2000). Each of these theories applied a specific view and modelling of the business domain focusing in activities and performance of (human) actions (TOA), human communications (EO) and context for actions together with the dependency between context and actions (OS). NOMIS takes some essential aspects of these views in its own vision and expands them in a more realistic, comprehensive and concise view of the IS reality.

NOMIS foundational theories share a key common element – the *human action* – as *the* driver of information systems. Actually, all things done or to be done within any IS are driven by human actions. Motivated and supported by HR, and central in the foundational theories, NOMIS central element is the *observable human action*. It is shown at the centre in the NOMIS vision presented in Figure 1.

NOMIS Vision, besides the *observable human action* central element and, the three views inspired by NOMIS foundational theories adds a fourth view – the information view – acknowledging the importance of information. The remaining elements from figure 1 are the human performer behind each action and *Norms*. Norms or *social norms* are a concept inherited from OS that is used to regulate human actions and provide a way to model expected behaviour.

Each view and the norms concept will be briefly described and explained in the next sub-sections. A deeper discussion on the presentation and analysis of the foundational theories of NOMIS and its vision can be found in Cordeiro et al, 2010.

2.2.1 The Interaction View

The Interaction View covers the communicational dimension of human action. All (human) interactions involve communication and communication itself is a form of interaction. Any business or organisation is driven by a network of people performing actions coordinated by communication; interactions link people. This perspective draws a special attention in how people interact and, in particular, communicate.

This view covers the IS modelling perspective of EO and extends it. It is not restricted by the single interaction pattern – the business transaction pattern – used in EO to model organisations.

In this view the different aspects involving interaction, such as who are the communicating actors, what interactions they perform, what communication links or channels connect them, and other observable aspects may be addressed and represented.

2.2.2 The State View

The state view looks into environmental conditions or states and their dependencies that enable a human agent to act. It is concerned with context, state and state dependencies related to human actions. This is an essential perspective behind OS. The environment, including the appropriate elements, enables or affords the human agent the ability to execute a certain action. These states are called in NOMIS *environmental states* (ES). ES usually represents essential business states that are related by existence dependencies to other ES. An ES can be composed by a single physical element (a *body*), or a single information element (an *information item*), or a group of different bodies and information elements in a particular state. The elements composing an ES have some observable form that may include information by referring to its physical representation.

2.2.3 The Physical View

The physical view is focused on the material and observable aspects related to human actions. This view covers the material dimension of human action expressed by TOA and addresses actions and their relationships to *bodies*: how *bodies* are affected and carried by actions. A particular representation under this view is the representation of business processes showing action sequences and activities. It is important to note that in NOMIS, business process models show different elements and follow different rules. As an example, in NOMIS vision, only human actions should be included, action sequence relates to expected human behaviour regulated by norms and the initial activity before entering an action and the final activity after leaving it represent states of the environment.

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

2.2.4 The Information View

The Information view covers the information dimension of human action. The importance of information is recognised by all information system theories and its significance to human action should be emphasised. Most of human actions depend or rely on information in different ways. Some of them cannot even be performed without it. Therefore the identification of the important information required for each action must take special attention. There are some assumptions NOMIS makes in alignment with HR and its foundational theories: (1) information does not exist without a material support: a body or a human actor and, (2) information is created by humans or special bodies known, in NOMIS, as *instruments* and can only be consumed 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. This means, an awareness system.

Information is also used by *norms* where it is related to agents and human actions. This is another responsibility assigned to the information view - to identify and represent the information needed by norms.

2.2.5 Norms

NOMIS views previously described provide a coherent and comprehensive view of IS centred in human action and information. Each of them offers a different perspective, however they are related in a coherent and consistent model of the IS. The elements shown and represented in each of them should be the same. A coordination act from the interaction view is a human action and can be used in all other views, the same should happen for any other human action, body, human performer and information item depicted in any view. Besides those connection points NOMIS uses also the OS *norm* concept to regulate human actions and provide a way to model expected behaviour. In this case, behavioural norms, which are related to human actions, are used. Behavioural norms are represented analytically in a semi-formal way as defined in OS as follows:

IF condition **THEN** agent
ADOPTS attitude **TOWARD** something

This form includes the key components of NOMIS: human performers as agents, human actions as attitudes and environmental states as conditions.

Besides regulating human behaviour, groups of norms are used to establish another kernel concept of NOMIS that is imported from OS - information fields (IF) – where specific terms and behaviours are understood in a similar way by the members of a community under those IF. This notion is used in NOMIS to define the boundaries of the terminology used in a particular IS.

2.3 NOMIS Models

Models, and modelling artefacts, are used to show simplified views of the reality, capturing its essential elements according to a particular ontology. This representation of the reality may be seen as a specific language and, as any language, determine the way world is perceived, the way plans are established and the way world is acted upon. Following these ideas, NOMIS Models define a new notation that includes a set of tables and diagrams to represent the different NOMIS views according to its Vision. Although there is a pre-set of representation artefacts suggested for each view these should not be seen as the only ones that can be used for that view and purpose. Besides the NOMIS notation also the Unified Modelling Language (UML) was extended with a set of dedicated profiles and can be used to represent NOMIS (Cordeiro and Liu, 2007; Cordeiro and Liu, 2008).

Because of space restrictions NOMIS model notation will not be described here, nevertheless examples will be given in the next section using UML profiles that will be fully explained.

3 MODELLING INFORMATION SYSTEMS WITH NOMIS

NOMIS Vision provided in the previous section establishes a new way of understanding and analysing IS. This Vision has consequences on the way we model the IS, some of these consequences are issues found in current modelling approaches, some other relate to innovative aspects not currently addressed. These innovative aspects will be the focus of this paper. In order to illustrate these aspects a simple use case of a library system will be used. A few parts of this use case will be modelled using NOMIS Models applying its UML profiles.

3.1 The Library System Use Case

The library system use case is described and proposed in Dietz, 2006 as a basis for modelling using the EO theory. It is enough for this paper to know the general idea behind it. It is just a library that lends books to their clients requiring, in this case, a membership. There are processes for lending and returning books, for applying for membership, including a special applying process for a reduced membership fee. The actors and the action and activities locations are also identified.

3.2 Applying NOMIS – Preliminary Steps

NOMIS Modelling approach does not propose a modelling methodology. Therefore, a preliminary step will be to analyse the problem and to extract its key elements according to NOMIS Vision and

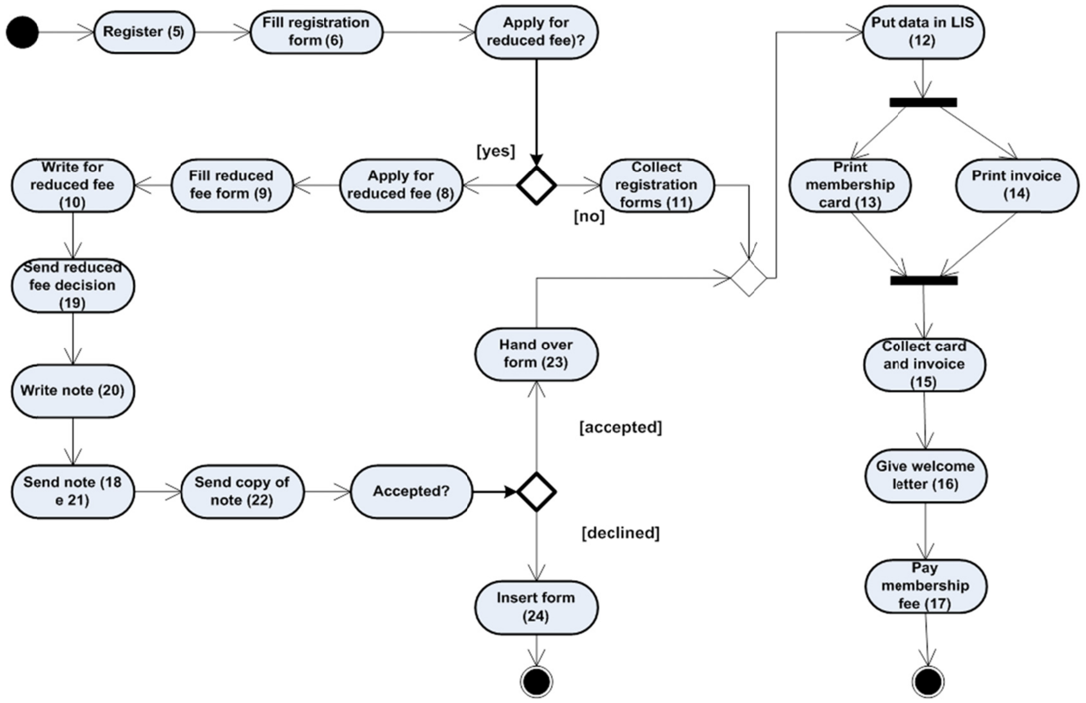


Figure 2: An Action Sequence Diagram of the library registering process.

modelling views. In this case two key elements that are also at the kernel of NOMIS views are human actions and their performers. These elements together with some other information will be collected using a special artefact named Human Action Table (HAT). HAT registers human actions, their related human actors, namely the initiator of that action and the addressee in case he/her exists, and also related action elements such as physical elements (*bodies*), *information items* and locations. Using HAT elements will facilitate the description of business processes that may be depicted with a NOMIS action sequence diagram (ASD). In figure 2 there is an example of an ASD describing the registering process required by a person to become a library member in our case study. A first distinction of NOMIS vision and other modelling approaches is that actions addressed and shown by NOMIS must be always *human observable actions*; Human actions clearly distinct what is done by computers from what is done by people. Additionally, only human actions, through their performers, have assigned responsibility, intention, commitment, and other important human and business aspects. Focusing on human actions also reveals the non-deterministic nature of the relationship between them. Effectively, action sequences determined by human actions are regulated by norms depending on people expected behaviour. Anyone following these norms may decide to break these rules.

3.3 States, Environmental States and the State View

The notion of a state is present in other modelling approaches but NOMIS defines it in a particular way. We will start with a simple *body state* that relates to a physical element that exhibits a specific condition. An example could be a form with two possible states filled and unfilled. Not all *body states* need to be addressed. From our library use case an important *body* with some relevant specific states is the *book*. A library book may be available for lend, lent or, otherwise, not available for lend as it was not registered in the library system. Each different state will allow different human actions. Figure 3 shows the book state diagram (similar to a UML state diagram). In this figure the letter I inside a circle means an *informational state*. This type of state cannot be extracted directly as it is defined by information recorded elsewhere. Although it seems a simple detail it is important. For example, if the "lent" information is recorded in a book using a label or something else this state could be extracted by anyone or any physical tool as it is *observable information*. The notion of *observability* from Human Relativism applied here is acting as a step

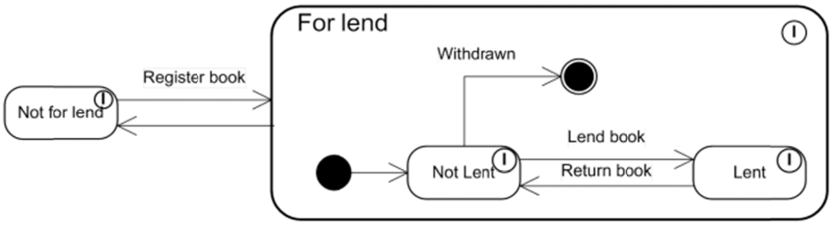


Figure 3: A Body State Diagram of a library book.

toward the desired IS modelling precision. NOMIS also defines human roles as specific (human) states acknowledging the physical (or *body*) nature of human beings.

NOMIS uses a broader notion of a state as well: the *Environmental State* (ES). An ES in NOMIS gathers all the necessary pre-conditions for a specific human action to occur. These conditions or states should be defined using NOMIS elements, namely bodies, information items and human performers. We think the idea behind ES is one of the most innovative ones created by NOMIS. An ES may be understood as defining a place from where paths are originated and end to. Paths are business processes (BP). An analogy with physics would relate ES to potential energy and BP to kinetic energy. This analogy may be used to highlight the instability problems that we have modelling BP. BP are subject to change, like trying to model or determine the trajectory of a projectile compared to the stability provided by the starting and target position. It is possible a projectile to reach the target from a wide variety of ways (BP) but the target position (ES) will stay mostly stable. In the library system case study, there are four important ES, namely book lent, book for lend, membership and paid membership fee. These ES depend on each other: a lent book needs a book for lend and a valid membership; a valid membership needs a library member with a paid fee. Figure 4 shows those ES together the dependencies between them represented with dashed arrows.

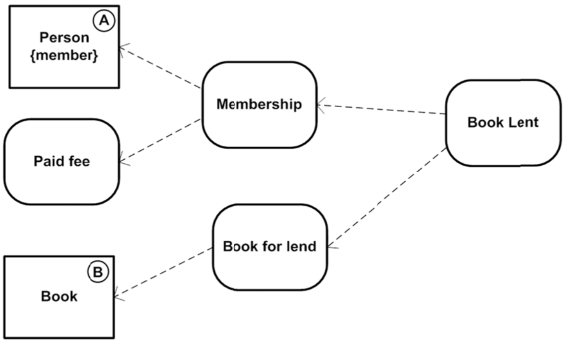


Figure 4: An Existential Dependency Diagram of the library system.

ES in NOMIS are not just concepts; they are completely defined using NOMIS elements as mentioned. A membership ES is composed by the library member, an *information item* with the member contacts, name, legal information and a paid fee information. It is relevant to observe that looking at the identified states, BPs used to reach them corresponds to main library system BPs. A membership ES is reached by following the “to register” process, the book lent ES is reached with the “to lend a book” process, the book for lent is reached by an ordering and registering the book process and so on. Any of these BP may be changed with time although most states will remain valid for the library system.

3.4 Library Action Views

NOMIS interaction and physical views are focused in human actions and their components. The first one deals with human communication and other forms of interactions and the second one with actions and participating bodies. Within these views all type of actions and their sequences can be modelled using ASD. Actions can also be grouped using activities that are, in NOMIS, *composite actions*. Some groups of actions can form patterns that can also be defined in NOMIS. Patterns usually have placeholders in the place of specific actions. These last aspects can be found in other modelling approaches. One interesting aspect of NOMIS is the ability to show action and activity patterns using templates.

Both the physical view and the interaction view use similar diagrams to represent action sequences that are similar to UML activity diagrams. In the case of the physical view these diagrams also include the representation of locations and actor roles as *swimlanes*, and *bodies* and its *states* as *object nodes*.

The interaction view also allows presenting interactions between actors by using a Human Interaction Diagram (HID). Using these diagrams permit us to acknowledge the different actions and

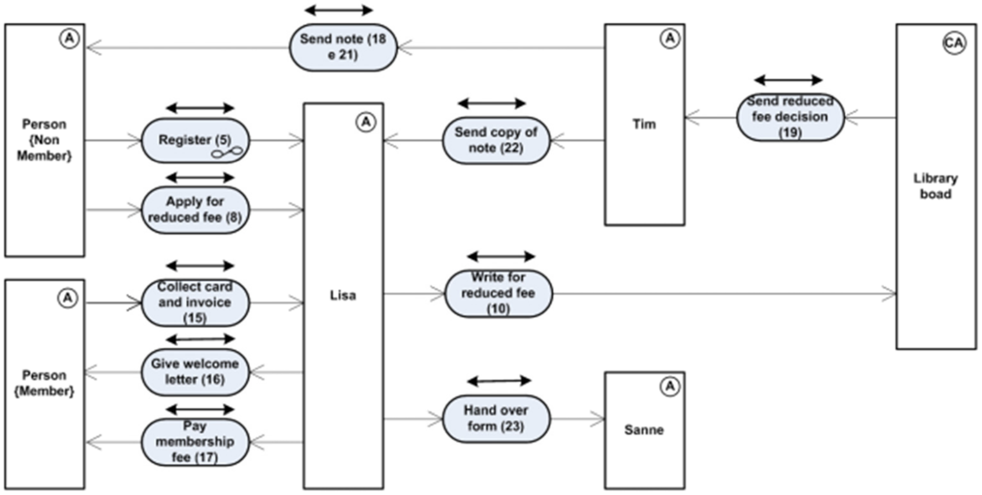


Figure 5: A Human Interaction Diagram of the library system.

activities where more than one person is involved. In figure 5 a HID of the library is presented. This diagram shows different interactions (with a double arrow on top) related to a single human action or a group of actions or interaction activities (activities have a special symbol inside, see (5) in the figure). This view highlights the important dimension of communication and collaboration sometimes missing or not emphasized in other modelling approaches.

The physical view has also another representation artefact, namely the Action View Diagram that is used to show individual actions together all related elements specifically its human performer, involved bodies and needed information items. This last element has a special relevance. The focus on human actions reveals us the need to support these actions and, information items, besides those directly used in the action, such as information on why, when, how to perform the action become important action helpers. For the information system being developed these helpers may also be part or constitute a dedicated awareness system.

3.5 Information and Norms

NOMIS information view goes a step further in information analysis. Following the general principles of HR, there are at least three things to note about information: 1) information cannot be seen independent from a human interpreter; 2) information is always carried by a physical element or *body*; and 3) information about a body may not be observable. Regarding 1) NOMIS has an Information Connection Diagram (ICD) where actions, information items and their supporting bodies are represented. In these diagrams is possible to show the information transferences between bodies. An interesting example is the library registering process. During this process the registration data is communicated by the future member to the librarian that introduces it in the library system. From the IS point of view there is a fundamental difference between being the library to introduce the information in the system or the applicant. In the first case the information item moves from the applicant to the librarian and then to the library system. This may lead to data errors during the transfer process, an example would be a misunderstanding of a particular data element by the librarian. The ICD would show this information transfer clearly. In 2) the highlight is the ability to have different bodies carrying the same information. Finally, 3) may be related to the issue identified before of some bodies having specific *information states* meaning that they are not *observable*. As an example a book in the library may not be for lend. This information cannot be extracted from the book if it is not observable. To distinguish between a body observable and not observable information makes an important difference.

NOMIS Norms relates information or environmental states to actions and their human performers. Business rules under NOMIS are mostly norms. NOMIS classifies Norms as:

1. **Requirement Norms**: needed for the action execution
2. **Trigger Norms**: action triggered given some information made available.

Table 1: Results of applying the CQF to the selected modelling techniques.

	NOMIS	DFD	EM	UML	RAD	BPMN
Organisational (Who)	(Human) Actor	----	----	Actor	Role	Participant
Functional (What)	Activity	Process diagrams	----	Use case	Process and roles	Process maps and descriptions
Behavioural (How)	Action	Processes	----	Action	Action interaction	Activities
Temporal (When)	Existential dependency	----	----	State event	State event	Events
Contextual (Where)	Environmental state / Information field	Context diagram	----	----	----	Pools and lanes
Motivational (Why)	----	----	----	----	State description (goal)	----
Resource (With)	Body	Data flow, data store	Entities	Object	----	Data Object
Conceptual (Which)	Information Item	----	Entity type	Classifier	----	----

3. **Information Norms**: as helpers or just informative for individual actions

Most norms are implicit and only those relevant to the system should be modelled.

As a final remark related to norms is that norms regulate expected human behaviour. They are a form of having some system stability although peopl behave according to different system of norms. As an example in our homes with our family we have a particular behaviour, surely not the same behaviour we adopt in our work organization. The same happens with different fields of knowledge; the terms and ontologies defined for libraries are different from those of medicine for example. Each system of norms defines a specific *Information Field* (IF). This notion used by NOMIS has its origin in Organisational Semiotics. The importance for modelling the library system is that we will find at least two different and simultaneous IF: the first one is the librarians that have their own terms and field knowledge and the second one is the public in general that lends books. NOMIS Vision calls our attention to these differences that should be taken care in IS modelling.

4 DISCUSSION AND RELATED WORK

In order to evaluate NOMIS modelling innovative aspects it will be useful to compare its key notions with other modelling approaches. First, it is important to know and understand how information systems development (ISD) uses or relies on modelling. In practice, modelling is commonly applied in ISD through visual representation techniques being diagrams the most common one. It is also common to differentiate between process modelling techniques and IS modelling techniques where process analysis and modelling is studied under the separate and related field of business process management and, data analysis and modelling together with object-oriented analysis and modelling remain within the information systems field. George Giaglis (2001), for example, propose a taxonomy of business process modelling and information systems modelling techniques where he studies and compares commonly used modelling techniques of both perspectives.

Typical dimensions for comparing modelling techniques are the functional, behavioural, informational and organisational perspectives (e.g. Giaglis, 2001; de Cesare and Serrano, 2006). However, for this comparison, it seems a better choice to use the Common Questions Framework (Cordeiro and Filipe, 2005) that covers all dimensions mentioned before and adds a few important ones. To stress the modelling differences between NOMIS and other modelling approaches we choose to select just a small group of well-known modelling techniques commonly used, namely:

- IS modelling techniques: Data Flow Diagrams (DFD), Entity Modelling (see Avison and Fitzgerald, 2006) and Unified Modelling Language (UML) (OMG UML, 2014)

- Business process modelling techniques: Role Activity Diagrams (RAD) (Ould, 1995; Ould, 2005), Business Process Modelling Notation (BPMN) (OMG BPMN, 2014).

The results of applying the CQF to these modelling techniques is shown in table 1. This table helps to have a broader view of the main aspects covered by each technique. In this case NOMIS furnishes a better coverage of the analysed dimensions together with UML and BPMN, although BPMN is more specific. It should be noted that many methodologies use more than one modelling technique (for example EM and DFD) to help them representing better the organisational domain. Nevertheless, this broader view does not really emphasize the benefits of using NOMIS, in this case it is necessary to look into a few particular aspects of NOMIS. To start with, the human centeredness of NOMIS realized by its exclusive modelling of *human observable actions* is one of the most important differences to other approaches. Anything done in a business or organisation is *always* done by humans through human actions. Any action performed by a machine becomes part of its technical implementation and at most, may only be judged and accepted in the business domain. Also, this focus on human action allows some key social aspects to be correctly addressed, such as responsibility, commitment, interests and intentions that have a strong emphasis in NOMIS foundational theories. The next difference is the use of context in NOMIS. In NOMIS context is addressed in different ways: firstly through the use of environmental states which are necessary conditions for a group of actions to be performed, secondly regarding each individual action by covering all elements, including information items related to that action and, thirdly, through information fields that provide a field of shared understanding for each IS used element. The analysed modelling techniques do not restrict actions to human actions mixing technical details with organisational elements and making difficult to understand and establish properly the organisational-technical boundary. Also these techniques do not make a proper context for understanding each model element. NOMIS does this by allowing each element to be perceived under an information field (IF) scope allowing for a similar element be understood differently under different IFs also represented in NOMIS models. Another difference is coherence and consistency as any NOMIS views *sees* the same reality where an element of one view is always related to the same element in another view. This cannot be done when a methodology chooses to use different modelling techniques for representing the same organisational reality. In this case UML may overcome this difficulty, however UML was not thought to model the business domain which results in having many problems to represent it besides mixing technical details and using an *object-oriented* paradigm (Avison and Fitzgerald, 2006). Another differentiating aspect of NOMIS is human communication that is not addressed by other modelling techniques despite its importance in organisations. In this case BMPN is an exception having a conversation diagram and recognizing the important of this organisational view. A last major distinct aspect in NOMIS is the notion, representation and use of norms. All analysed modelling techniques use a sequential or parallel flow for actions, but they do not recognise it as being human dependent and possibly not followed (also, because they use the same flow for machine actions). In NOMIS the use of norms recognizes properly the human dependent nature of human action process flows.

5 CONCLUSIONS

This paper briefly presented NOMIS – a new modelling approach for information systems that integrates the theories of Organisational Semiotics, Enterprise Ontology and the Theory of Organized Activity. NOMIS is strongly founded in the philosophical stance of Human Relativism that is expected to provide the necessary modelling precision. NOMIS is fully described in (Cordeiro, 2011).

The focus of this paper was to show some important innovative modelling aspects of NOMIS that may be useful for a better understanding of IS and business systems. These aspects were also compared within different dimensions with similar concepts used by a small set of other well-known modelling techniques.

REFERENCES

Avison, D. and Fitzgerald, G., (2006). *Information Systems Development: Methodologies, Techniques and Tools, 4th Ed*, McGraw-Hill Education, UK.

de Cesare, S. and Serrano, A., (2006). Collaborative Modeling Using UML and Business Process Simulation. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, Vol. 1.

Cordeiro, J. (2011). *Normative Approach to Information systems Modelling*. PhD Thesis. The University of Reading, UK.

Cordeiro, J. and Filipe, J., (2005). Comparative Analysis of Ontology Charting with other Modelling Techniques. In *Proceedings of the 8th International Workshop on Organisational Semiotics*. Toulouse.

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.

Cordeiro, J., Filipe, J. and Liu, K., (2010). NOMIS – A Human Centred Modelling Approach of Information Systems. In *Proceedings of the 4th International Workshop on Enterprise Systems and Technology*. Athens, Greece.

Dietz, J., (2006). *Enterprise Ontology, Theory and Methodology*. Springer-Verlag, Berlin Heidelberg, Germany.

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.

Giaglis, G., (2001). A Taxonomy of Business Process Modelling and Information Systems Modeling Techniques, In *International Journal of Flexible Manufacturing Systems*, 13, pages 209-228.

Holt, A., (1997). *Organized Activity and Its Support by Computer*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Liu, K., (2000). *Semiotics in Information Systems Engineering*. Cambridge University Press, Cambridge, UK.

OMG BPMN [online], (2014). Object Management Group / Business Process Modelling Notation. Available from: http://www.bpmn.org [2014].

OMG UML [online], (2014). Object Management Group Unified Modelling Language. Available from: http://www.uml.org [2014].

Ould, M. A., (1995). Business Processes – Modelling and Analysis for Re-Engineering and Improvement. John Wiley & Sons, Chichester, England.

Ould, M. A., (2005). Business Process Management – A Rigorous Approach. The British Computer Society, Wiltshire, UK.

Sauer, C., (1997). Deciding the Future for IS Failures: Not the Choice You Might Think. In *Re-Thinking Management Information Systems*. Galliers, R.D., & Currie, W. (eds)., Oxford University Press.

An Approach to the Context-oriented Use Case Analysis

Kalinka Kaloyanova[1] and Neli Maneva[2]
[1]*Faculty of Mathematics and Informatics, Sofia University, 5 J. Bourchier Blvd., Sofia, Bulgaria*
[2]*Institute of Mathematics and Informatics, BAS, Acad. G. Bonchev Str., Bl.8, Sofia, Bulgaria*
kkaloyanova@fmi.uni-sofia.bg, neman@math.bas.bg

Keywords: Requirements Engineering, Use Case Analysis, Information Systems Development, Multiple-criteria Decision Making, Comparative Analysis.

Abstract: The paper describes our efforts to propose a feasible solution of a significant problem in information systems development – requirements engineering, based on use cases. The adjustable use case quality model is constructed and used within the Comparative Analysis method to support the decision making during the use case analysis process. Two real-life problems related to this process are described and their solutions through the suggested approach are given.

1 INTRODUCTION

Nowadays the volumes of information are continually increasing, the methods for storing and searching information become more complex and the software users - more demanding. One way to meet their expectations is to assure an effective engineering of the elucidated user-defined requirements. Requirements Engineering (RE) as a subfield of both Information Systems (IS) and Software Engineering (SE) concerns the real-life needs of the users and the constraints for the systems to be built.

Recognizing the importance of the RE in any software intensive system's life cycle and understanding the significant role of the requirements elicitation activity, we address a challenging issue – providing systematic approaches to the construction and constantly improvement of the requirements for information systems development.

The goal of this paper is to present our approach to IS requirements engineering, based on quality use cases. The applicability of a formal method for the proposed context-oriented use case analysis is investigated in Section 2. The most difficult activity – use case quality modelling in a specified context, is described in Section 3. Two examples are given, illustrating how the method can be successfully applied. In the Conclusions section some directions for future research and experimental work are shared.

2 BASIC DEFINITIONS AND RELATED APPROACHES

Requirements engineering is one of the initial and very important activities in software development. Lately its significance is realized and there are plenty of methods, frameworks, tools, etc., supporting RE (Pressman, 2009), (Sommerville, 2011), (Denny, 2005). From practitioner's point of view the great variety of approaches and available tools are more distractive than useful. This is especially true for information systems development, where the main requirements come from end users and usually are ambiguous, contradictory and incomplete (Pokorny, 2010). Due to such reasons the iterative approaches usually are more effective, because they support the requirements refinement in a more constructive way (Kaloyanova, 2012).

2.1 Use Case Modelling

A common form for describing functional requirements for a system is the ***use case modelling*** (Maciaszek, 2005)**.** It captures user requirements of a new system by detailing all scenarios that the users will perform. Each use case describes a specific way of using the designed system by actors – users with specific roles, trying to achieve their goals (Sinnig, 2010). The set of all use cases defined by actors describes the system functionality (Cockburn, 2000).

Use case modelling usually is recognized as a part of Requirement engineering discipline of the Unified Process (UP) that is focused primarily on the functional requirements of the system (Kruchten, 2004). The non-functional requirements are divided by the UP approach into several categories. The acronym FURPS+ presents the functional and several important non-functional requirements, which are quite relevant to IS development – usability, reliability, performance, and supportability of the system (Larman, 2004).

The use case model describes not only the system behaviour, but it also defines the boundary of the system and how this system communicates with different actors.

Besides, as use case modelling is a part of the UP, starting from the first phase of it - Inception, the use cases capture functionality of the system in an iterative process that is not only understandable for the users, but could also be detailed in the next phases of the UP – Elaboration and Construction.

Use cases are described in natural language. So they can be easily understood by non-technical people and can be further discussed by users, clients, analysts, designers, developers and their managers.

The use cases can be written with different level of details (Larman, 2004):

- Brief – one-paragraph summary, focused on the main success scenario;
- Casual – multiple paragraphs, covered various scenarios of user-system interaction;
- Fully dressed – all scenarios are written in detail, comprising some supporting sections, such as preconditions, post-conditions, special requirements, etc.

The different forms of use cases description can be created during successive iterations, following the UP. In this way, at every step some new information is added to the use case description. The analysis usually starts with a set of use cases in brief format and detail them through the next iterations. In order to do this systematically, the use cases have to be ranked, taking into account their significance, risk, complexity or other criterion of prioritization.

Several questions arise here - how to implement the use case definition method in practice and how to evaluate the quality of use case description in different iterations. The analysts need a technique that supports a stable evolution of the specifications of every element of software functionality, described by the corresponding use case.

A number of works discuss different sides of the use case development – formalization (Sinning, 2009), content (Cockburn, 2000), inspection techniques (Anda, 2002), etc. Most of the research publications about the use case modelling are summarized in (Torner, 2006). Although the intensive research in the area it is hard to find a method that represents a holistic view on all aspects of the use case transformation.

In this paper we introduce a use case quality model for the description of the functionality, presented at different consecutive iterations of the same use case. Further this quality model can be built into a method allowing not only to assess the achieved use case quality, but to facilitate some decisions, made during the process of quality improvement. We base our approach on the formal Comparative Analysis method (Maneva, 2007). This method was chosen primarily because it shares the main objectives and methods of the broad theory of the Multiple Criteria Decision Making, trying to specify and apply them systematically.

2.2 Comparative Analysis

We will present the essence of the **Comparative Analysis (CA)** in order to explain how it can be used for the activity under consideration – use case analysis.

Comparative Analysis is a study of the quality content of a set of homogeneous objects and their mutual comparison in order to select the best, to rank them (establishing a preference order) or to classify each object to one of the predefined quality categories.

For CA use we distinguish two main roles: the ***CA Analyst***, responsible for all aspects of CA implementation, and a ***CA customer*** – a single person or a group of individuals, tasked with making a decision in a given situation. Depending on the identified problem to be solved at a given moment, a ***case*** should be opened to determine the context of the desired Comparative Analysis.

Each case is specified by the following elements:

case={View, Goal, Object, Competitors, Task, Level }

The **View** describes the CA customer's role and the perspective from which the CA will be performed.

The **Goal** expresses the main customer's intentions in CA accomplishment such as to describe, analyse, estimate, improve, predict or any other, formulated by the Customer, defining the case.

The **Object** represents the item under consideration. In the field of software engineering any studied object belongs to one of the following groups: products, processes or resources. For each object for CA application, a quality model should be created – a set of characteristics, selected to represent the quality content in this context, and the relationships among them.

According to the goal, the set C of **Competitors, C = {C_1, C_2, ... , C_n}** – the instances of the objects to be compared – should be chosen.

The element **Task** of a case can be ***Selection*** (finding the best), ***Ranking*** (producing an ordered list), ***Classification*** (splitting the competitors to a few preliminary defined quality groups) or any combination of them.

The parameter **Level** defines the overall complexity of the CA and depends on the importance of the problem under consideration and on the resources needed for CA implementation.

Usually the object quality model is presented as a hierarchy. At the top of the hierarchy is the total object quality. The first level comprises some user-oriented attributes, called ***factors***. The next level describes a number of object-dependent attributes, providing quality. These ***criteria*** can be further decomposed to more simple and measurable characteristics. To each node at the hierarchy structure a weight (a coefficient of importance) is assigned, and for the leaves of the hierarchy some appropriate ***metrics*** are defined. Starting a bottom-up evaluation of characteristics at each level in the hierarchy and applying a modification of the MECCA (Multi-Element Component Comparison and Analysis) method, described in (Bowen, 1985), we can obtain the quantitative measures of all factors and fill the competitors-factors matrix **E(nxm)**, where **n** is the number of the competitors, defined in the set **C**, and **m** is the number of the quality factors. Each element **Ei,j** is the measure of the **i**-th competitor with respect to the **j**-th quality factor. The obtained matrix **E** is further used as input to the software tools, implementing the required selection, ranking or classification methods.

Additional details about the CA method can be found in a number of recent publications. In this paper we provide only the information, absolutely necessary for understanding the CA application as a method, supporting decision making.

3 A CONTEXT-ORIENTED USE CASE ANALYSIS

The CA method can be used in any decision making situation within the use case analysis, after specifying its context.

According to the definition already given in Section 2.2, the concrete situation for a particular decision making can be stated by a case, comprising six elements:

case={View, Goal, Object, Competitors, Task, Level}

The ***View*** describes the CA customer's role and focuses on the perspective of the Comparative Analysis. Taking into account the responsibilities and typical tasks of the main participants in the use case analysis, the following Customer's roles have been identified till now: ***Business Analyst, User, IS developer.*** Thus a lot of situations can be further specified, reflecting the specific participant's point of view to the analysed context.

As the ***Goal*** expresses the main Customer's intentions in CA accomplishment and bearing in mind the goal-oriented use case definition and its further improvement, the Analyst should decompose the goal stated for use case analysis to a number of CA-related goals, e.g. to describe, assess, estimate, improve, predict, etc.

The element **Object**, as it was mentioned above, represents the item under consideration. Particularly for the use case analysis, the following objects, classified in three groups, can be studied:

Products – use case, collection of use cases, use cases source (discussion memos, documentation, specifications, user's and developer's stories), prototypes for checking selected use case characteristics, like completeness, usefulness, etc.

Processes, related to use case analysis: elicitation, quality assessment, prioritization, documenting, maintenance, tracking, etc.

Resources, related to use case analysis: technological (e.g. method, technique, tool), or project-oriented (people, team, performance).

For each studied object a quality model should be created – a set of characteristics, selected to represent the quality content, and the relationships among them.

According to the goal, the set **C** of **Competitors** – the instances of the objects to be mutually studied and compared – should be chosen. If the Goal is only to create the quality model of the object under consideration, the set C is empty.

There are no special considerations, when we define the element ***Task*** of a situation for use case analysis. As usual, the task can be *Selection*, *Ranking*, *Classification* or any combination of them.

It is not difficult to define the degree of complexity (simple, medium or high) presented by the element **Level.**

One of the challenges in the implementation of the CA is the construction of a model, adequate to the quality content of the object, pointed in a given case. According to the prescriptions, the Analyst is responsible for accomplishment of this task, which is with very high cognitive complexity and usually requires unified efforts of the Analyst and the Customer, ordered the CA.

3.1 A Quality Model for Object "Use Case"

As an example, we will describe the basic hierarchical model, constructed for the quality of object "use case". In this model we propose three user-oriented characteristics (factors) – Applicability, Validity and Utility.

The factor ***Applicability*** concerns the relation between the use case and the real system functionality and addresses the question of the use case legality. Next follows a brief definition of the quality characteristics selected for this factor and the corresponding metrics for the leaves of the tree structure.

The hierarchy obtained after the decomposition of the ***Applicability*** factor is shown in Table 1.

The first level of the hierarchy comprises three criteria affecting the factor **Applicability** - ***Accuracy, Trackability,*** and ***Adequacy.***

Table 1: Decomposition of the factor Applicability.

Factor Applicability	
Criteria	Item-Oriented Characteristics
Accuracy	
Trackability	
Adequacy	Robustness Completeness Correctness

Accuracy defines how the use case corresponds to the reality. It shows whether the goal of this use case could be reached by the presented use case description. Accuracy can be measured by the ratio of correct features to the total number of features, described in the use case.

Trackability reflects whether the changes in reality can be accompanied by changes in the use case.

The ***Adequacy*** criterion is more complex**.** It should be evaluated by answering several important questions like:

- Is the use case robust?
- Is the use case complete? Are all possible scenarios defined?
- Is the use case correct? Is the right sequence of steps described in the scenarios?

In the frame of ***Adequacy*** the characteristic **Correctness** can be measured by the ratio of correct steps to the total number of steps, described in the use case. More sophisticated evaluation of the correctness can be done by splitting this characteristic into two more – *Compliance* and *Homogeneity,* defined as follows:

– *Compliance*: Does the use case description follow the proper separation between the positive scenario and other alternative scenarios?

– *Homogeneity:* Are the descriptions of all scenarios with the same level of details?

The second suggested factor ***Validity*** concerns the inner capacity of the use case to produce results corresponding to the expected ones.

Table 2: Decomposition of the factor Validity.

Factor Validity	
Criteria	Item-Oriented Characteristics
Reliability	
Modifiability	Complexity Modularity
Understandability	Self-descriptive Concise Structured
Measurability	

Table 2 illustrates the proposed hierarchy for this factor based on the selected quality criteria: ***Reliability, Modifiability, Understandability and Measurability***.

The quality criterion ***Reliability*** is evaluated through the answers of two questions:

- Is the use case adequate?
- Is the use case robust?

The next criterion ***Modifiability*** shows how easily the use case can be modified. It is decomposed further in two measurable characteristics:

- *Complexity* – can be considered as textual or algorithmic, measured by the appropriately constructed metrics;

- *Modularity* – usually measured by the number of all alternative scenarios.

Nevertheless the quality criterion ***Understandability*** is well known concept, for the use case analysis it concerns variety of details, which can be taken into account:

- Is the use case self-descriptive? Are all steps clear to follow?
- Is the use case concise? Is the description of the steps brief and clear?
- Is the use case well structured? Do all scenarios fit each other? Is the main positive scenario properly defined and all alternative scenarios are adequate to it?

The used metrics here reflect the structure – *linear* (usually for the brief form of the use case) or *hierarchical* (for the description of the alternative scenarios and their connections to the steps of the main positive scenario).

When we consider the criterion ***Measurability***, it is obligatory to select some concrete metrics for use case assessment like:

- The number of steps of the successfull scenario;
- The number of steps, performed by each actor, involved into the success scenario;
- The number of system steps, described by the success scenario;
- The number of all alternative scenarios;
- The number of steps of alternative scenarios;
- The number of steps of the negative alternative scenarios;
- The number of all steps, described in the studied use case;
- The number of preconditions in the studied use case.

The third factor ***Utility*** deals with the relation between use case and users and addresses the question of the possibility to derive conclusions from the created use case. The proposed hierarchy for this factor is shown in Table 3, presenting a number of quality characteristics.

Table 3: Decomposition of the factor Utility.

Factor Utility	
Criteria	Item-Oriented Characteristics
Content	
Fidelity	
Constructiveness	
Stability	
Usability	Effectiveness Efficiency User satisfaction

The ***Content*** criterion focuses on the text description. Several questions arise here:

What does the use case represent?

Is use case detailed enough? Are all steps clear and described with appropriate level of detail?

The criterion ***Fidelity*** answers the question if different users will get similar results using the same use case.

The next criterion - ***Constructiveness*** explains how the use case facilitates some future IS-related activities like design, coding, testing, etc.

The criterion ***Stability*** explains if the use case can be manipulated to obtain false results or if the new version of the use case corrupts any of the previous versions.

The last considered criterion ***Usability*** can be evaluated from three different perspectives:

- *Effectiveness:* Does the use case match the purpose? Are all steps of the main positive scenarios leading to the goal?

- *Efficiency*: Does the use case fulfil the stated goal with minimal resources? Is the sequence of the steps is the shortest way to the goal?

- *User satisfaction:* Can be measured by the percentage of unsatisfied users and the average rating, given by users.

3.2 A Quality Model for Object “A Set of Use Cases”

This object can appear in a situation involving a set **U** of interconnected use cases, which can be studied as a whole due to some customer’s considerations. For this object a simple linear model can be created, comprising four quality characteristics - ***Completeness***, ***Consistency, Relevance,*** and ***Correctness.***

For the purposes of the use case analysis we define their meaning as follows:

***Completeness* –** the set **U** comprises all developed use cases, necessary to described the functionality of a system or a system’s compound element – component, subsystem, etc.;

***Consistency* –** all involved in **U** use cases are in reasonable and logical harmony, without any contradictions in their content;

Relevance* –** all involved in **U** use cases should possess a direct and clearly identified connection to the studied system’s functionality. For this factor two measures have been applied – ***Recall and ***Precision***:

Recall measures how exhaustive a description of the scenarios is. It can be calculated by the following formula:

***Recall* = x / (x+y) * 100%,** where

- x is the number of described in **U** relevant scenarios,

- y is the number of the relevant, but not described yet in **U** scenarios.

Precision measures the amount of noise in scenarios description, based on the formula:

***Precision* = x / (x+z)** * 100%, where

- x is the number of described in **U** relevant scenarios;

- z is the number of described in **U**, but non-relevant scenarios.

The last quality characteristic **–** ***Correctness,*** is related to some scenarios errors in **U**, which have been found and fixed during the use case analysis. It can be measured by the ratio of corrected scenarios to the number of all scenarios described in the set **U.**

4 EXAMPLES

The usefulness and feasibility of our idea to apply the CA method to use case analysis have been examined for solving two real-life problems, described below.

Case Study 1: How the CA can Support the Iterative Use Case Development. The iterative use case development is based on defining the initial version of the use case and then - a repetitive process of use case quality assessment and further refinement. According to the results of assessment, next iteration of the use case should be created through some changes that improve the recent quality content and functionality of the use case.

The CA with properly defined context can be used as a form of research for communicating and evolving a use case, as consecutive versions of the use case are implemented within the following step-wise procedure:

Step 1. ***Pre-analysis*** – description of a number of real-life problems encountered during the activity under consideration, for which the CA method seems to be useful. During the iterative use case development some additional information about the current state should be gathered to decide how to continue. From CA perspective this can be done as follows:

- Creating a derivative quality model of the use case, including only those quality characteristics from the generic model, which are relevant to the considered context;
- Evaluating the current version of a use case from the point of view of different actors;
- Comparing two consecutive iterations of the same use case to observe the effect of the performed changes on the quality;
- Evaluating the last iteration of a use case to decide how to proceed. In this situation the use case estimates are compared with those of a virtual "perfect" use case, whose current state is described by the same quality characteristics, but with preliminary assigned threshold values.

Step 2. ***Preparation*** – defining the CA context and planning the CA implementation. The elements (View, Goal, Object, Competitors, Task and Level) of each case must be specified. The relevant sources of information, needed for the CA performance, are identified and made available. A CA plan is created, describing some parameters of the work – duration, cost, personnel involved, tasks and responsibilities allocation and schedule.

For the above mentioned activities the possible elements of a case can be:

View – that of any already identified participant in use case analysis - user, IS developer or business analyst;

Goal – to assess, to compare or any other, defined by the CA Customer;

Object – a use case;

Competitors – an instance of a use case (and a "perfect" use case for the last action for comparison of the current use case with the "perfect" one);

Task – usually it is ranking;

Level – simple, middle or high.

Step 3. Construction – building a quality model, corresponding to use cases quality content for a defined case.

Step 4. ***Execution*** – evaluating the quality factors and accomplishment of the CA Task.

As an example, let us describe this step in CA for the case, designed for comparison of the last use case version and the "perfect" use case. The final results of the bottom up evaluation are presented in an objects-factors matrix. In it the first row comprises the threshold values, assigned to the perfect use case, and the second row comprises the results of use case evaluation (see Table 4).

Table 4: A filled objects-factors matrix.

Use case/ Factors	F1 Applicability weight - 0.3	F2 Validity weight-02	F3 Utility weight-05
Perfect use case	1	1	1
Investigated use case	0.8	0.7	1

Step 5. ***Completion*** - analysis, interpretation and drawing conclusions from the results.

In the discussed case the deviation of the estimated measures from the desired values for each factor are calculated and described, in order to make decision what to do further. The first possibility is to continue with new iterations, trying to improve the values for the second and the third factor. In this case we have to define the scope and contents of some additional use case refining activities. The second possibility is to accept the achieved quality level as appropriate and to stop analysing this use case.

Case Study 2: How the CA Can Support the Use Case Prioritization. One of the most difficult problems in use case analysis is to create an ordered list of all identified as important use cases. Such prioritization will make possible to perform the use case analysis in a more systematic and efficient way, facilitating the proper distribution of the planned and usually insufficient resources.

A Real-life Problem: After the use case quality content assessment is done, it is necessary to assign a priority to each use case in a given set of already developed use cases.

Some cases for this problem can be defined as a combination of the following elements:

View – can be that of the User, Business Analyst, Project manager or IS Developer;

Goal – to compare the use cases, selected for prioritization;

Object **–** use case;

Competitors – all developed use cases, chosen to be prioritized;

Task – ranking to produce an ordered list of the compared use cases;

Level – simple, middle or high.

5 CONCLUSIONS

The main purpose of this paper is to propose a systematic approach to the creation and continuous improvement of use cases, based on the Comparative Analysis method. It supports the decision making in some significant situations for the use case analysis.

The basic advantage of the CA is the possibility for adjustment to the context, specified by the elements of the investigated case. The most difficult activity – object quality modelling – has been described and illustrated with models for two basic objects – use case and a set of use cases. As example, three CA successful implementations in the field of use case analysis have been given.

Some possible directions for future work can be:

- To identify and describe entirely some other CA situations within the use case analysis, so as to enrich the collection of the re-used items – situations, models, metrics, etc.;
- To examine the possibility to apply the CA to other use case related activities as validating, tracking, change management, design and implementation of already defined use cases.

ACKNOWLEDGEMENTS

This work is partly supported by the National Scientific Research Fund under the Contract ДТК 02-69/2009 and partially supported by Sofia University "St. Kl. Ohridski" SRF/2014 under the Contract "Successful Practices for Information Systems Analysis and Design".

REFERENCES

Anda, B., D. Sjøberg, 2002. Towards an Inspection Technique for Use Case Models, In *Proceedings of the 14th Int. Conf. on Software Engineering and Knowledge Engineering*, pp. 127-134.

Bowen, T., G.B. Wigle, J. Tsai, 1985. Specification of software quality attributes, *Software quality evaluation guidebook,* RADC-TR-85-37, vol.III.

Cockburn, A., 2000. *Writing Effective Use Cases*, Addison-Wesley.

Denny, R., 2005. *Succeeding with Use Cases: Working Smart to Deliver Quality*, Addison Wesley.

Kaloyanova, K., 2012. Design from Data: How To Use Requirements for Better IS Analysis and Design, *Proceedings of the Int. Conference Informatics in Scientific Knowledge*, pp. 189-197.

Kruchten, P., 2004. *The Rational Unified Process: An Introduction*, Pearson Education.

Larman, G., 2004. *Applying UML and Patterns: An Introduction to Object-Oriented analysis and Design and Iterative Development*, 3rd Edition, Prentice Hall.

Maciaszek, L., 2005. *Requirements Analysis and System Design*, Addison-Wesley Longman, Inc.

Maneva, N., 2007. Comparative Analysis: A Feasible Software Engineering Method, *Serdica J. of Computing, 1(1), pp. 1-12.*

Pokorny, J. at all, 2010. *Information Systems Development: Business Systems and Services: Modelling and Development*, Springer.

Pressman, R., 2009. *Software Engineering: A Practitioner's Approach*, 7th Edition, McGraw-Hill.

Sinnig, D., P. Chalin., F Khendek, 2009, LTS Semantics for Use Case Models, In *Proceedings of the 2009 ACM symposium on Applied Computing*, pp. 365-370.

Sinnig, D., H. Javahery, 2010. Mastering Use Cases: Capturing Functional Requirements for Interactive Applications, In *Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems*, pp. 373-374.

Sommerville, I., 2011. *Software Engineering*, 9th Edition, Addison Wesley.

Tornew, F., M. Ivarsson, F. Pettersson, P. Ohman, 2006. Defect in automotive use cases, In *Proc. of 2006 ACM/IEEE international symposium on Empirical software engineering*, pp.115-123.

Validating Value Network Business Models by Ontologies

José Granjo[1,] Marzieh Bakhshandeh[1,2], João Pombinho[1], Miguel Mira da Silva[1] and Artur Caetano[1,2]
[1]*Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais 1, 1049-001 Lisbon, Portugal*
[2]*Information Systems Group, INESC-ID,Rua Alves Redol 9, 1000-029 Lisboa, Portugal*
{jose.granjo, marzieh.bakhshandeh, jpmp, mms, artur.caetano}@tecnico.ulisboa.pt

Keywords: Business Model Canvas, e3value, ArchiMate, Ontology Integration, Meta-model, OWL, SPARQL.

Abstract: Different meta-models allow modeling the business of an organization from different perspectives. The Business Model Canvas focus is close to the strategy of the organization. E3value allows modeling of value networks and ArchiMate allows alignment from business models to IT infrastructure. When models of these three meta-models coexist for a certain value network, they must be consistent. Currently, there is no way to validate such consistency automatically. We propose a solution, using ontologies and ontology mapping techniques (OWL, OWL.DL, SPARQL) that helps to validate instantiated models automatically, based on a set of mapping rules between the three meta-models. In this work, the mappings between Business Model Canvas, e3value and ArchiMate are identified and formalized through ontologies. The formalized mapping is then applied to a case study and exploited, together with reasoning techniques.

1 INTRODUCTION

Innovative business models challenge the traditionally established ways of generating value, resulting in advantage to a company. We have seen, over time, that innovative business models can dare the subsistence of other established companies or even create complete new markets.

Having a shared understanding of what is the business model of a company, by representing it, eliminates possible interpretation ambiguities.

The Business Model Canvas (BMC) (Osterwalder & Pigneur, 2010) is a tool for representing the business model of a company. When a company is executing its business, it is part of a network of companies that exchange value with the final goal of delivering value to customers. e3value (Gordijn & Akkermans, 2003) allows modeling of value networks: the value exchanges between actors in the network.

BMC allows representing the business model of an organization on a higher-level (or strategic) perspective. e3value is closer to operationalization of business, by showing value transactions of the value network. On a lower-level, business processes can be modeled with ArchiMate (The Open Group, 2012), a service-oriented enterprise architecture modeling language, which considers three different layers: business, application and technology. Toghether, these three meta-models allow the alignment from business models to information technology and infrastructure.

When modeling the business of an organization and its value network, several of these models can be instantiated. If theys coexist for a value network, they must be consistent and there is no way to automatically validate such consistency between model components. We aim to analyze the possibility to perform this validation between models by using ontologies. An ontology is a formal, explicit specification of a shared conceptualization (Gruber & others, 1993). Such search for inconsistencies helps business to IT alignment.

In section 2, the research proposal is presented, and next, in section 3, we reference each meta-model. Afterwards (section 4), the mapping rules between the three meta-models are presented. In section 5, a validation has been done with an example case study. Finally, conclusions and future work are discussed.

2 RESEARCH PROPOSAL

As depicted in figure 1, a validation method for models of BMC, e3value and ArchiMate is proposed, based on a set of mapping rules between the three meta-models and validation of those rules using reasoning techniques.

A formal representation is desirable in order to improve the meta-models conformance verification of models, through the verification of logical inconsistencies present in models. To define mapping rules between the meta-models is also required, in order to establish an alignment between the concepts of the meta-models that can assess the consistency between them. This proposal has a unified meta-model for the purpose of integration. Also, BMC, e3value and ArchiMate models will be instantiated in the integrated ontology. Ontology reasoning techniques will validate the correctness of the integrated ontology.

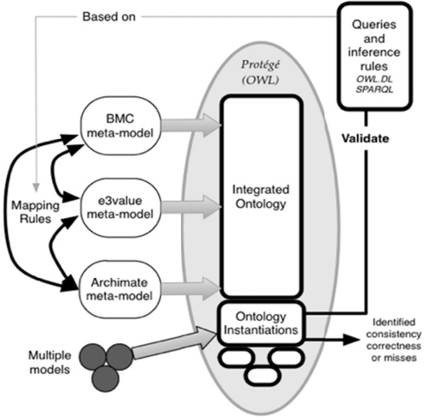


Figure 1: Research Proposal.

3 META-MODELS

In this section, the three meta-models used in this work (BMC, e3value, ArchiMate) are introduced.

The Business Model Canvas (BMC) (Osterwalder & Pigneur, 2010) aims to represent a business model and translate it into explicit knowledge by considering nine building blocks in a canvas. Its focus is close to a strategic perspective. BMC is based on the Business Model Ontology (Osterwalder & others, 2004).

E3value (Gordijn & Akkermans, 2003) enables value network modeling, aiming to provide a common understanding of a business idea executed by a network of actors that jointly create, distribute and consume value. A meta-model for e3value can be found in (Pombinho J. A., 2014).

ArchiMate is an open and independent modeling language, from the Open Group (The Open Group, 2012), The ArchiMate framework organizes its meta-model in a three by three matrix: the rows capture the domain layers (business, application, and technology); the columns capture cross layer aspects (active structure, behavior and passive structure).

4 MAPPING RULES

To define the mapping rules, the meta-model mapping technique defined in (Zivkovic, et al., 2007) was partially applied. It distinguishes different types of mappings. On one level, we consider class-to-class mappings, which define relations between concepts of two distinct meta-models. On another level, we consider equivalence and aggregation mappings between concepts. The referenced technique is only used to define the mappings since we do not aim to merge ontologies.

4.1 Ontology Integration

The word integration has been used with different meanings in the ontology field. In simple terms, ontology integration is the process of identifying common concepts and relationships shared between ontologies, (Sofia & Martins, 1999) (Euzenat, et al., 2007). Three main techniques of ontology integration are categorized as:

- **Ontology Alignment**: the process of building a new ontology by identifying correspondences between all the concepts of two ontologies,.
- **Ontology Mapping**: the process of building a new ontology by finding common concepts between two (or more) concepts belonging to two (or more) different ontologies.
- **Ontology Merging**: the process of building a new ontology by merging several ontologies into a single one that will "unify" all of them.

Different types of mismatches may occur between different ontologies (Davies, et al., 2006) (Kotis, et al., 2006) (Amrouch & Mostefai, 2012) (Bouquet, et al., 2004):

- **Syntactic Mismatches:** if different languages represent ontologies.
- **Lexical Mismatches:** heterogeneities in names of entities, instances, properties or relations.
- **Semantic Mismatches:** classified into three abstract forms: (1) coverage, two ontologies that cover different (possibly overlapping) portions of the world (or even of a single domain); (2) granularity, two ontologies where one provides a more/less detailed description of the same entity; (3) perspective, two ontologies where one provides a viewpoint on some domain,

which is different from the viewpoint adopted in another ontology.

For mapping BMC, e3value and ArchiMate, there were two kinds of mismatches: lexical mismatch, where the same entity is represented by two different names, such as, Customer Segments and Business Actor; and coverage mismatch, where from the same point of view, in the same context and with comparable vocabulary, part of the domain that is described differs and there are only overlapping parts (Value Proposition and Goal). Most ontology mapping approaches focus on automating the discovery of a mappings. This case, requires an exact mapping, so the mappings were done manually using (Zivkovic, et al., 2007).

4.2 Mapping BMC to e3value

Previous work (Gordijn, et al., 2005) shows connections between concepts of BMC and e3value to understand similarities and differences between both ontologies to possibly integrate them in order to improve representation, design and analysis of business models. The defined mapping rules (*table 1*) are inspired on previous work.

4.3 Mapping BMC to ArchiMate

Another work (Meertens, et al., 2012) explored the connection between BMC an ArchiMate, where the concepts of BMC were successfully mapped to e3value. The defined mapping rules (*table 2*) are inspired on previous work. We do not consider any mapping between Customer Relationships (CR) and Business Collaboration because CR refers to the types of relationships an organization maintains with its customers. Key Partners is only a list of partners, so the mapping is simplified to Business Actor. Cost Structure is not mapped to Value because it is only the cost of performing Key Activities and maintaining Key Resources.

4.4 Mapping e3value to ArchiMate

Direct transformation from e3value to ArchiMate is inhibited by different levels of abstraction between the economic transactions modeled in e3value and ArchiMate (de Kinderen, et al., 2012). The same authors use DEMO (Dietz, 2006) as a bridge for the different levels of abstraction of e3value and ArchiMate (de Kinderen, et al., 2012). Another work (Pombinho J. A., 2014) defines the mapping between e3value and DEMO in a more grounded, formal and thorough way. Namely, it specifies a detailed mapping based on the coordination acts and facts of the transactional pattern and the corresponding competences by the value actors. Additionally, the authors define a Value-oriented Solution Development Process in (Pombinho, 2013) that specifies a process for incrementally developing value networks by alternating coherent value and construction models. Table 3 shows the defined mapping rules.

Table 1: BMC-e3value meta-model concepts mapping.

BMC concept	E3value concept	Mapping rationale
Customer Segment	Actor	**Equivalence**. The Customer Segments are groups of people that a company aims to reach, while Actor is an independent economic entity that generates profit or increases its utility. (1:1)
	Market Segment	Analogous to Actor. Market Segment is a specialization of Actor.
Key Partner	Actor	**Equivalence**. Key Partners is the group of partners that help the businesses execution. Analogous mapping to Customer Segment. (1:1)
Channel	Value Transmission	**Aggregation.** A value transmission can be the delivery of value to customers through a certain channel. (many:1)
Key Activity	Value Activity	**Equivalence**. Key Activities are the most important things a company must do to make its business model work, while an actor performs a Value Activity for profit or to increase its utility. (1:many)
	Value Transmission	**Aggregation**. A Key Activity can involve a value exchange (to obtain a needed resource) with a Key Partner. (many:1)
Key Resources	Value Object	**Equivalence**. A Key Resource acquired from a Key Partner. (1:1)
Revenue Stream	Value Transmission	**Equivalence** to inbound and monetary value exchange. (1:1)
Value Proposition	Value Interface	**Equivalence**. A value interface defines the group of value objects the company is willing to provide. Those value objects are also defined in the outbound value ports belonging to the value interface. (1:1)
Actor	Actor	**Equivalence.** The Actor concept is the owner of a BMC. (1:1)

Table 2: BMC-e3value meta-model concepts mapping.

BMC concept	ArchiMate concept	Mapping rationale
Customer Segment	Business Actor	**Equivalence.** Customer Segments are groups of people that a company aims to reach, while Business Actor is an organizational entity that is capable of performing behavior. (1:1)
Key Partner	Business Actor	**Equivalence.** Key Partners is the group of partners that help the business model execution. Analogous to Customer Segments. (1:1)
Channel	Business Interface	**Equivalence.** Channels describe how a company communicates with and reaches its Customer Segments to deliver Value Propositions. A Business Interface is a point of access where a business service is made available to the environment. (1:1)
Revenue Stream	Value	**Equivalence.** Value may apply to what a party gets by selling or making available some product or service, or it may apply to what a party gets by buying or obtaining access to it. (1:1)
Value Proposition	Business Service	**Aggregation.** A Value proposition is a Business Service or a Product (1:many)
	Product	
	Value	**Aggregation.** The worth of the Service/Product for the Customer.
	Goal (Motivation Extension)	**Aggregation.** Why the Service/Product is useful for the Customer.
Key Activity	Business Interaction	**Equivalence.** The performed Key Activities may be represented as high-level Business Processes or Business Functions, or by Business Interactions between internal Business Actors. (1:many)
	Business Function	
	Business Process	
Actor	Business Actor	**Equivalence.** Analogous to Customer Segments, for example. (1:1)

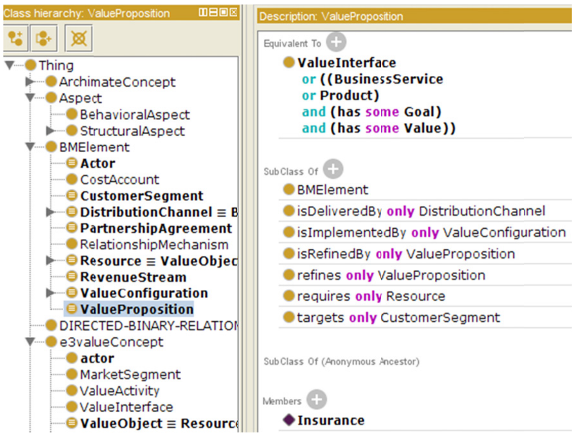
Table 3: e3value-Archimate meta-model concepts mapping.

E3value concept	ArchiMate concept	Mapping rationale
Actor	Business Actor	**Equivalence**. Actor is an independent economic entity that generates profit or increases its utility. Business actor is an organizational entity that is capable of performing behavior. (1:1)
Market Segment	Business Actor	**Equivalence**. Market Segment is a specialization of Actor. (1:1)
Value Interface	Product	**Equivalence.** A value interface groups the value objects offering provided by one actor. Such value offering in concretized by business services and a Product is a coherent grouping of business services. (1:1)
Value Transmission	Business Service	**Equivalence.** The utilization of a business service by an external actor is concretizes a value transmission. (1:1)
Value Activity	Business Process	**Equivalence.** High-level business processes that support business services offered to external business actors. Business process choreography is only present in lower levels.
Value Object	Business Object	**Equivalence.** A value object is a business object transmitted to some other actor. A business object is tangible. (1:many)
	Value	**Equivalence.** Value is the worth of a business service or product to some business actor. Value can represent intangible value objects. (1:many)

5 VALIDATION

This proposal has a unified meta-model for the purpose of integration. It was required to transform the three meta-models into ontology (OWL). The BMC OWL representation was obtained from other authors (Pigneur, 2004). The ArchiMate transformation process uses (1) an OWL representation of the ArchiMate meta-model and (2) OWL representations of ArchiMate models (Bakhshadeh, et al., 2014) (Antunes, et al., 2013) (Bakhshandeh, et al., 2013). An e3value OWL representation was implemented with inspiration on the meta-model presented in (Pombinho J. A., 2014).

Figure 2, shows a partial of the integrated ontology, along with relationships with other concepts and some constrains. It was required to instantiate the models inside the integrated ontology as individuals (OWL). A transformation was made from BMC, e3value and ArchiMate example models to individuals.


Class hierarchy: ValueProposition
Description: ValueProposition
Equivalent To
SubClass Of
SubClass Of (Anonymous Ancestor)
Members

```
Thing
  ArchimateConcept
  Aspect
    BehavioralAspect
    StructuralAspect
  BMElement
    Actor
    CostAccount
    CustomerSegment
    DistributionChannel ≡ B
    PartnershipAgreement
    RelationshipMechanism
    Resource ≡ ValueObjec
    RevenueStream
    ValueConfiguration
    ValueProposition
  DIRECTED-BINARY-RELATIOI
  e3valueConcept
    actor
    MarketSegment
    ValueActivity
    ValueInterface
    ValueObject ≡ Resourc
```

```
Equivalent To
ValueInterface
 or ((BusinessService
 or Product)
 and (has some Goal)
 and (has some Value))

SubClass Of
BMElement
isDeliveredBy only DistributionChannel
isImplementedBy only ValueConfiguration
isRefinedBy only ValueProposition
refines only ValueProposition
requires only Resource
targets only CustomerSegment

SubClass Of (Anonymous Ancestor)

Members
Insurance
```

Figure 2: Partial OWL representation of the integrated ontology.

5.1 Case Study

ArchiSurance example models were used as a case study to analyze the current ontology. Example models have been taken from (The Open Group, 2012) (Meertens, et al., 2012), except for the e3value, in which a coherent simple model has been designed for the purpose.

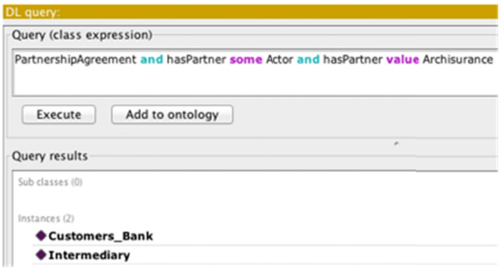
The example models have been converted to the OWL to be represented as the instances of the integrated ontology. The example in this section is used to illustrate the capabilities of reasoning, by validating the correctness of the integrated ontology. A set of predefined competency questions (Fox & Gruninger, 1998) were used in order to validate ontology.

5.2 Reasoning

In the recent years logical reasoning has been widely used in the field of ontology engineering (Baader, et al., 2008) (Corcho, et al., 2006) (Lenzerini, et al., 2004). The set of competency questions defined to validate the integrated ontology is composed by the following questions:

1. What are the Value Propositions that have Value “Be_Insured”?
2. What are the Key Partners of “Archisurance”?
3. What are the Value Propositions that have the Goal “Reduce_maintenance_costs”?

In figure 3 to 5, such questions can be formalized into a description logic queries. The first two queries are in OWL-DL and the last one in SPARQL .

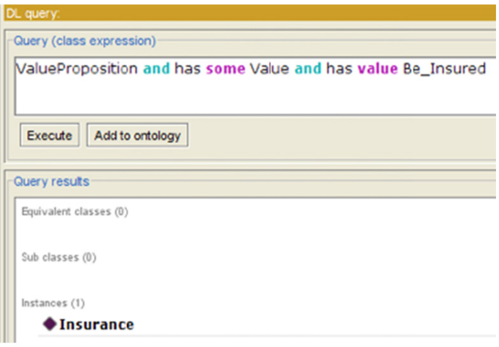


```
PartnershipAgreement and hasPartner some Actor and hasPartner value Archisurance

Sub classes (0)

Instances (2)
Customers_Bank
Intermediary
```

Figure 3: Query1 result.



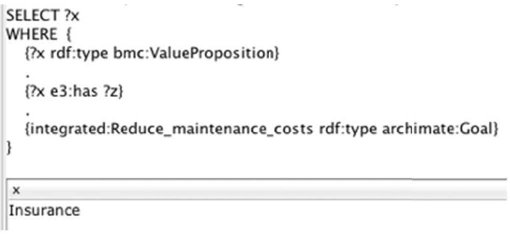
```
ValueProposition and has some Value and has value Be_Insured

Equivalent classes (0)

Sub classes (0)

Instances (1)
Insurance
```

Figure 4: Query2 result.



```
SELECT ?x
WHERE {
   {?x rdf:type bmc:ValueProposition}
   .
   {?x e3:has ?z}
   .
   {integrated:Reduce_maintenance_costs rdf:type archimate:Goal}
}

x
Insurance
```

Figure 5: Query4 result.

6 CONCLUSIONS

In this paper, we showed how to bridge between BMC, ArchiMate and e3value meta-models along with their models, by the use of ontologies and ontology mapping techniques.

An integrated ontology was created from the three meta-model ontologies and example models were instantiated into it. This integrated ontology was validated through a case study via logical reasoning techniques. The linkage between the three meta-models allows automatic consistency validation of models of the three meta-models.

Future work will focus on the application of this approach to new scenarios in order to explore the analysis possibilities, considering the usage of different reasoning and querying techniques.

ACKNOWLEDGEMENTS

This project is partially supported by the European Commission under the 7th Framework Programme (FP7/2007-2013) under grant agreement 269940, TIMBUS project (http://timbusproject.net). Also, by national funds through FCT – Fundação para a Ciência e a Tecnologia, under project PEst-OE/EEI/LA0021/2013.

REFERENCES

Amrouch, S. & Mostefai, S., 2012. *Survey on the literature of ontology mapping, alignment and merging.* s.l., s.n., pp. 1-5.

Antunes, G. et al., 2013. *Using ontologies for enterprise architecture model alignment.* s.l., s.n.

Baader, F., Horrocks, I. & Sattler, U., 2008. Description logics. *Foundations of Artificial Intelligence,* Volume 3, pp. 135-179.

Bakhshadeh, M., Morais, A., Caetano, A. & Borbinha, J., 2014. Ontology Transformation of Enterprise Architecture Models. In: *Technological Innovation for Collective Awareness Systems.* s.l.:Springer, pp. 55-62.

Bakhshandeh, M. et al., 2013. *A Modular Ontology for the Enterprise Architecture Domain.* s.l., s.n., pp. 5-12.

Bouquet, P. et al., 2004. D2. 2.1 Specification of a common framework for characterizing alignment.

Corcho, O., Fernández-López, M. & Gómez-Pérez, A., 2006. Ontological engineering: principles, methods, tools and languages. In: *Ontologies for software engineering and software technology.* s.l.:Springer, pp. 1-48.

Davies, J., Studer, R. & Warren, P., 2006. *Semantic Web technologies: trends and research in ontology-based systems.* s.l.:Wiley. com.

de Kinderen, S., Gaaloul, K. & Proper, H. A., 2012. Bridging value modelling to ArchiMate via transaction modelling. *Software & Systems Modeling,* pp. 1-15.

de Kinderen, S., Gaaloul, K. & Proper, H. E., 2012. Integrating value modelling into archimate. In: *Exploring Services Science.* s.l.:Springer, pp. 125-139.

Dietz, J. L. G., 2006. *Enterprise Ontology: Theory and Methodology.* s.l.:Springer.

Euzenat, J., Euzenat, J., Shvaiko, P. & others, 2007. *Ontology matching.* s.l.:Springer.

Fox, M. S. & Gruninger, M., 1998. Enterprise modeling. *AI magazine,* 19(3), p. 109.

Gordijn, J. & Akkermans, J., 2003. Value-based requirements engineering: exploring innovative e-commerce ideas. *Requirements engineering,* 8(2), pp. 114-134.

Gordijn, J., Osterwalder, A. & Pigneur, Y., 2005. Comparing two business model ontologies for designing e-business models and value constellations. *Proceedings of the 18th Bled eConference, Bled, Slovenia,* pp. 6-8.

Gruber, T. R. & others, 1993. A translation approach to portable ontology specifications. *Knowledge acquisition,* 5(2), pp. 199-220.

Kotis, K., Vouros, G. A. & Stergiou, K., 2006. Towards automatic merging of domain ontologies: The HCONE-merge approach. *Web Semantics: Science, Services and Agents on the World Wide Web,* 4(1), pp. 60-79.

Lenzerini, M., Milano, D. & Poggi, A., 2004. Ontology representation & reasoning. *Universit di Roma La Sapienza, Roma, Italy, Tech. Rep. NoE InterOp (IST-508011).*

Meertens, L. O. et al., 2012. *Mapping the business model canvas to ArchiMate.* s.l., s.n., pp. 1694-1701.

Osterwalder, A. & others, 2004. The business model ontology: A proposition in a design science approach. *Institut d'Informatique et Organisation. Lausanne, Switzerland, University of Lausanne, Ecole des Hautes Etudes Commerciales HEC,* Volume 173.

Osterwalder, A. & Pigneur, Y., 2010. *Business model generation: a handbook for visionaries, game changers, and challengers.* s.l.:John Wiley & Sons.

Pigneur, Y., 2004. *Modeling the business model ontology with Protégé and OWL,* s.l.: s.n.

Pombinho, J. A. D., T. J., 2013. *Value-oriented Solution Development Process - uncovering the rationale behind organization components..* s.l., s.n.

Pombinho, J. A. D., T. J., 2014. *Linking Value Chains – Combining e3Value and DEMO for specifying Value Networks.* s.l., s.n.

Sofia, A. G.-P. P. H. & Martins, J. P., 1999. *Some issues on ontology integration.* s.l., s.n.

The Open Group, 2012. *ArchiMate 2.0 Specification.* s.l.:s.n.

Zivkovic, S., Kuhn, H. & Karagiannis, D., 2007. *Facilitate Modelling Using Method Integration: An Approach Using Mappings and Integration Rules.* s.l., s.n., pp. 2038-2049.

Actors Based Competences Supporting Enterprise Modeling Changes

Marwen Jabloun, Yemna Sayeb and Henda Ben Ghezala
Riadi–GDL Laboratory, Manouba University, Manouba, Tunis, Tunisia
marwenjabloun@gmail.com, yemna.sayeb@gmail.com, hhbg.hhbg@gnet.tn

Khaled Gaaloul
Public Research Centre Henri Tudor, Luxembourg
khaled.gaaloul@tudor.lu

Keywords: Enterprise Architecture, Actor, Competency, Performance, Ontology.

Abstract: The competitive environment of companies triggers changes having impacts on the enterprise information system. In this context, Enterprise Architecture (EA) offers instruments to steer enterprises changes and to ensure cohesion and alignment between the different aspects of an enterprise. Despite considerable numbers of EA frameworks, there are not many giving importance to actors competencies supporting EA changes when modeling. In this paper, we present an approach based on competencies to support enterprises' changes. In doing so, we introduce a performance assessment method from the process modeling to the competence selection using a competence-oriented ontology. Moreover, we develop a prototype as an evaluation part of our work illustrated by a case study supporting EA decision making.

1 INTRODUCTION

The growing importance of alignment, compliance and manageability issues have increased the attention to EA (Braun et al., 2005). EA promises to better align technical projects with business needs, where business values could best be realized by a holistic approach to systems architecture that explicitly looked at every important issue from every important perspective (Zachman, 1987).

EA presents methodology for guiding changes. Those changes can impact different levels of the enterprise. This needs an approach steering EA changes. More specifically, the organizational aspect is more and more considered where actors, roles and competencies have a fundamental contribution in the success or the failure of desired changes (Gagnon, 2008). Enterprise performance depends on the alignment of the corporate strategy and business process. The performance of business process depends on the involved actors. These actors have skills to achieve their tasks using relevant IT systems (CIGREF, 2008). Hence, any EA change should take into account actor's competencies to assess its performance. The importance of actor and competency concept is clearly shown in enterprise models while ignoring change issues (Le Boulaire et al., 2008).

In this paper, we aim to focus on competencies management supporting EA change. Our approach is based, firstly, on a BPEL (business process execution language) process that optimizes EA change scenarios. The BPEL process ensures the orchestration of services using business rules from our performance assessment method. The performance assessment integrates, then, a competence-oriented ontology to ensure capturing relevant information for actors' performance (e.g., skills, training, etc.). Finally, we develop a prototype as an evaluation part of our work illustrated by a case study supporting EA decision making.

The remainder of this paper is structured as follows. Section 2 introduces the research context and problem statements. The approach is presented in section 3, where we explain our methodology to monitor process execution based a competence-oriented ontology. Section 4 evaluates our work illustrated by a real world scenario. Section 5 concludes and discusses future works.

2 BACKGROUND

2.1 Related Work

Traditionally, improvement initiatives are currently positioned at two types of performance improvement: continuous improvement and radical improvement. EA is a continuous improvement approach. The improvement is one of the basics of quality managed by a set of standard such as the ISO 9000 that describes which requirements a system needs for quality management. The ISO 9001 standard emphasizes the continuous improvement of the overall performance of the company. This principle means that management must measure performance based on quality (Hachicha, 2012).

Competence management aims to improve the enterprise performance through the effective deployment of resources allocated to the business process. Competence management involves the following steps: (i) Analysis of existing data; (ii) Analysis of decision change, and (iii) The design of the new organization.

Several meanings and definitions of *competence* exist, but no common definition seems to satisfy all viewpoints (Pepiot, 2012). In fact, there is an evolution in the considerations of the competence concept in enterprise modeling languages. The Unified Enterprise Modeling Language (UEML) proposes a consensus in the scientific community both at the terminology and conceptual (Petit et al., 2002). The actor concept in UEML is considered as a resource which is a specialization of the enterprise object. Resources can be specialized into three categories: applications, human resources and machines. UEML project has open research perspective. There are investigations of complementary language for competencies modeling such us the Unified Enterprise Competence Modeling Language (UECML). UECML is an enterprise modeling approach based on competences. UECML is based on a set of core constructs and a set of additional constructs. These additional constructs are specialized constructs required by the competence and resource based on enterprise modeling needs (Pepiot, 2012). The actor here is considered as an enterprise object. It is a subclass of enterprise object class. Therefore, it inherits all the properties of objects organization. The actor is assigned to a role and is characterized by a set of competence. We notice three granularity levels related to competence: (1) Unit competence which is considered as the basic level and entity in the model is tightly linked to an activity; it is defined as the ability to mobilize effectively material and non-material resources in order to respond to an activity; (2) Individual competence is the set of unit competences and resources developed/required by an actor within the framework of assigned activities; (3) Collective competences is considered as the highest competence level and linked to processes and group of actors.

Another aspect is about the link between competence and performance. In literature, many researches develop the competence integration in performance assessment. In (Bennour et al., 2006) authors proposed to model and assess the competence based on knowledge, rather than an overall assessment of competence. In (Hlaoittinun et al., 2009), they quantify the impact of both individual core competencies (technical, decision autonomy) and collective (management, relational) to assess the modulation rate of the nominal performance to execute an activity. Finally, the approach proposed in (Gruatet al., 2006) introduced the level of competency where a performance is calculated based on the operator's productivity such as in production line.

2.2 Research Motivations

Despite the competence extension, UECML is still lacking of a systematic approach evaluating competences based on resources. In a context where actors vary from the manager who makes strategic decision to the worker who performs operational activities, the issue is to ensure consistency with regards to actor's understandings and viewpoints. We think that leveraging competence features in EA can ensure this consistency. To that end, we propose a performance method that predicts impacts based on actor's competence. Prediction will allow architects to deal with situations such as the lack of resources when an actor is absent or has no skills to execute an activity with the process. Based on our approach, we can offer alternatives as anticipatory actions such as training, outsourcing to solve such problems.

3 THE APPROACH

The performance assessment based on competence is realized in three main steps: (1) Identification of KPIs and metrics derived from EA strategy in order to deal with the normative dimension of performance assessment; (2) Collecting relevant data relevant for competence computation where we propose an ontology to deal with cognitive dimension; (3) Development of business rules in order to orchestrate services invocations from the BPEL process.

The normative dimension is based on a set of KPI an d corresponding metrics. It provides metrics for business rules to assess performance. From the literature (Rezaei et al., 2011), we have identified the different services: developing a performance measurement system needs the conversion of strategic objectives into key goals, the establishment of metrics to compare the desired performance with the actual achieved standards, the gaps identification to allow us to understand performance and finally the initiation of improvement actions.

The trend in performance assessment is geared towards intangible success factors such as competencies, customer satisfaction, motivation and staff satisfaction (Rezaei et al., 2011). In our approach, we are based on competence concept because it is interrelated within EA layers, and so, provides a broad vision that facilitates decision making. Moreover, KPIs depend on the competence family. In fact, according to (CGIREF, 2009) competencies related to design, operation, infrastructure and technical architecture tend to move away from the enterprise core and even to move geographically away.

Performance assessment may have some semantic issues when dealing with metrics interoperability due to cultural differences between actors, e.g., business people and information technology professionals (Frank, 2002). It is a critical problem because it may leads to data collection aberrations.

Ontologies have been considered as solution to ensure the metrics interoperability in process reengineering (Charlet et al., 2002). We define an ontology entitled Ontology of Enterprise Oriented Competence (OnEOC) (see figure 1). The OnEOC ontology is composed of the following components:

- Thing, an abstract object representing the class of all things.
- Urbanization view refers to the different EA views: strategy view, the business view, the organizational view and the technical view.
- Work unit is a component that refers to a decision Unit it is linked with the performance estimation and the choice of changes reasons.
- The actor concept is a key concept. Actors have different competencies and roles. The actor in OnEOC can be considered through different views depending on his role and his capabilities.
- The competence concept identifies the unitary competence, personal competence and collective competence.
- The change concept is also defined in the OnEOC where it ensures the evolution of an existing state (AS-IS) to a desired target state (TO-BE).

The OnEOC ontology allows different types of research via ontology queries. For instance, actors executing tasks with a given KPI value such as the number of resolved complaints in an hour. The ontology allows a semantic interpretation of KPI, where technical actors interprets the KPI query execution time by the computer system, while actors at the strategic level interprets this KPI according to customers satisfaction percentage. Then, after identifying KPI and collecting relevant data; the performance assessment needs to develop business rules. Every business rules depends on the identified KPI and the extracted data from the ontology. These business rules will be executed via a process implemented by the business process execution language (BPEL). The idea is to allow combining a set of services for specifying business rules where performances are evaluated by orchestrating these rules.

BPEL is an XML language designed to allow the composition and orchestration of web services. BPEL inputs translate the different possible changes scenarios and the output is a value of the expected performance after change. BPEL inputs define the subject (reason) of the desired change, e.g., the stakeholder management to modify existing business processes. Changes criteria are either immediate or continuous. Change's nature can be related to specific actions such as add, delete or modify involving actors, systems, and processes. Then depending on the input, an invocation of a web service is executed based on business rules. The principle of performance assessment depends on the identification of the gap between existing competences and required competences allowing the adaptation to a given change. So the output of t this process represents the value of performance after change. Note that depending on the importance of the impacted actor and the importance of the impacted business domain, a decision can be taken in order to plan appropriate actions to deal with any performance lowering.

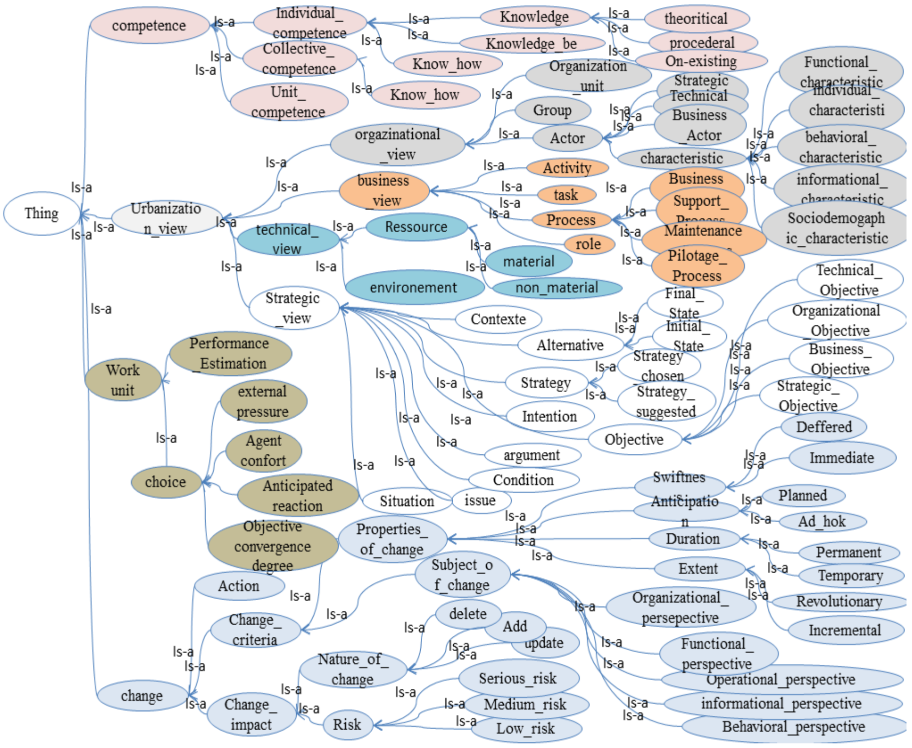


Figure 1: Ontology of Enterprise Oriented Competence (OnEOC).

4 EXPERIMENTATION

To illustrate and validate the proposed approach, we present here a case study of a Telecom operator. We have implemented the liaison between the ontology, the BPEL process and its business rules. The synchronized services by BPEL are deployed as web services. They are defined as requests using SPARQL queries (SPARQL Protocol and RDF Query Language) (Perez et al., 2009). These SPARQL queries are used to express the business rules implemented in Java classes.

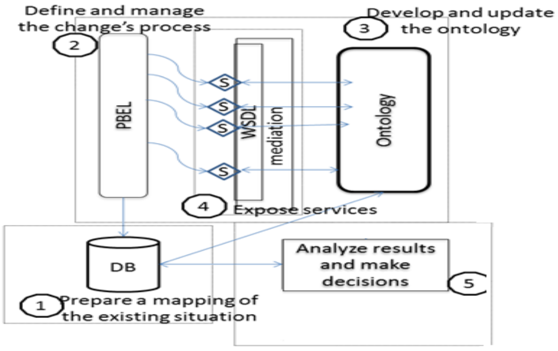

Define and manage the change's process
2
PBEL
S
S
S
S
WSDL
mediation
Develop and update the ontology
3
Ontology
4
Expose services
DB
1
Prepare a mapping of the existing situation
Analyze results and make decisions
5

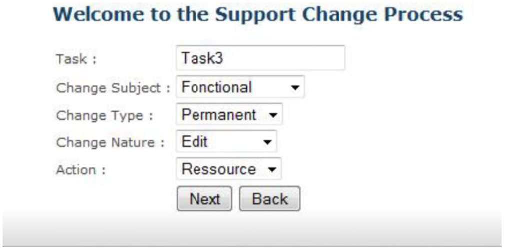
Figure 2: Prototype supporting process change.

Figure 2 depicts the overall architecture of the implemented prototype where the main components are presented by rectangles. The headed-arrow connections define the interaction between components and lozenges represent web services.

The Implemented BPEL process defines the execution of invoked processes by the web application. It is composed of many portion separated by an else/if condition. This condition depends on change's subject. Each process's portion calls the following operations: *Receive* to get the input, *Assign* to map the process input data and the Web service input data, *Invoke* to calls a web service, *Assign* to map the process output data and the Web service output data, and *Reply* to return results.

The proposed prototype is composed of a web application supporting a web interface that allows user to monitor change. The user can define the type, the subject and the nature of the change scenario via a web form. The web application offers also an interpretation to show the returned results.

In figure 3, we can change the actor assigned to task 3. This change can be permanent, temporary or immediate. The change may include additional actions (e.g., add, delete, modify).



Welcome to the Support Change Process

Task : Task3

Change Subject : Fonctional

Change Type : Permanent

Change Nature : Edit

Action : Ressource

Figure 3: Change process interface.

5 CONCLUSIONS

The main contribution of this paper is to support enterprise changes based on the competence concept. This concept contributes in performance assessment to monitor changes and support decision-making in EA. Supporting change process using competence is supported by the OnEOC ontology of competence addressing semantic issues when defining metrics for actors' performance.

We aim to improve this approach by proposing a method for modeling business rules. The idea is to help business experts to model the desired business domain in a natural language supported by a vocabulary that supply the necessary semantics to describe the overall framework. Standards like Semantics of Business Vocabulary and Business Rules (SBVR) and the Semantic Web Rule Language (SWRL) will be our next step to rationalize actors' performance in EA frameworks.

REFERENCES

Braun, C., Winter, R., 2005. *A Comprehensive Enterprise Architecture Metamodel and Its Implementation Using a Metamodeling Platform*. Institute for Information Management University of St. Gallen, Switzerland.

Zachman, J.A., 1987. *A Framework for Information Systems Architecture*. IBM Systems Journal, Volume 26.

Gagnon, Y-C., 2008. Thèse: *Les trois leviers stratégiques de la réussite du changement technologique*. École nationale d'administration publique du Québec.

CIGREF livre blanc, 2008. *L'Architecture d'Entreprise Un cadre global de coopération pour les acteurs de l'entreprise.*

Le Boulaire, M., Retour, D., 2008. *Gestion des compétences, stratégie et performance de l'entreprise :quel est le rôle de la fonction RH ?*. CERAG et Institut d'Administration des Entreprises de Grenoble.

Charpon, J., 2006. Thèse: *L'urbanisme organisationnel : méthode et aide à la décision pour piloter l'évolution du système d'information de l'entreprise*. L'Ecole Nationale Supérieure des Mines de Saint Etienne et de l'Université Jean Monnet.

Hachicha, M., 2012. Thèse: *Contribution à la Modélisation et Résolution du Problème d'Affectation sous Contraintes de Compétences et Préférences.* UNIVERSITE PARIS 8 Vincennes-Saint-Denis.

Petit, M., Doumeingts, G., 2002. *Report on the State of the Art in Enterprise Modeling*, Projet UEML: Unified Enterprise Modeling Language.

Pépio, G., Cheikhrouhou, N., Furbringer, M., and Glardon, R., 2006. *UECML: Unified Enterprise Competence Modeling Language.* Ecole Polytechnique Fédérale de Lausanne, Laboratory for Production Management and Processes, Lausanne, Switzerland.

Bennour, M. and Crestani, D., 2006. *Using competencies in performance estimation: From the activity to the process*. Université Montpellier 2, Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier.

Hlaoittinun, O., 2009. Thèse : *Contribution à la construction d'équipes de conception couplant la structuration du projet et le pilotage des compétences*. Université Franche-Comté, France.

Gruat, F.A., Botta-Genoulaz, V., Campagne, J.P., 2006. *Modélisation d'un problème d'ordonnancement avec prise en compte des compétences.* MOSIM'06, Rabat, Maroc.

Rezaei, A.R., Çelik, T., Baalousha, Y., 2011. *Performance measurement in a quality management system.* Department of Civil Engineering, Eastern Mediterranean University.

CIGREF, 2008. Livre blanc sur l'évolution des compétences les Systèmes d'Information, École de Management des Systèmes d'Information de Grenoble

Frank, U., 2002. *Multi-Perspective Enterprise Modeling (MEMO) - Conceptual Framework and Modeling Languages*, Proceedings of the Hawaii International Conference on System Sciences.

Charlet, J., Bachimont, B., Troncy, R., 2002. *Ontologies pour le Web sémantique*. Mission de recherche STIM, Institut National Université Technologique de Compiègne.

Perez, J., Arenas, M., Gutierrez, C., 2009. *Semantics and Complexity of SPARQL*. ACM Trans. Database Syst. ACM, New York, USA.

Towards a Generic Data Model for REA Based Applications

Bernhard Wally and Christian Huemer

Institute of Software Technology & Interactive Systems, Vienna University of Technology, Favoritenstr. 9-11, Vienna, Austria
wally@big.tuwien.ac.at, huemer@big.tuwien.ac.at

Keywords: Software Engineering, Business Models, Business Ontologies, Database Engineering.

Abstract: The original REA accounting model (McCarthy, 1982) has been extended in previous years into a business modeling language. Apart from its conceptual model, academic effort has been put into the definition of a formal description language, based on standards such as UML or OWL. The specification of a generic data model from a software engineering point of view for domain independent use of REA for business model specification and execution has been touched only briefly in the past. Thus, we present a data model concept for runtime-configurable REA business model definition and execution.

1 INTRODUCTION

The REA accounting model (McCarthy, 1982) was initially developed as a modern accounting methodology based on real world artefacts instead of the rather artificial double entry book keeping, originating in the Middle Ages. Over the years, it has been extended towards a business modeling language, which enables the description of value chains and enterprise resource planning (Geerts and McCarthy, 1997; Geerts and McCarthy, 2000; Gailly and Poels, 2007).

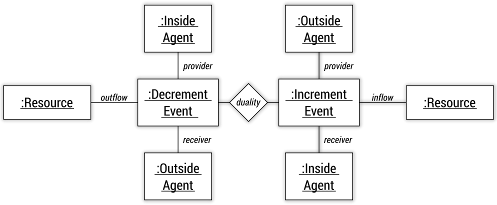


Figure 1: Sample object diagram of REA's core concepts.

Figure 1 depicts the REA core concept in terms of an UML object diagram: an economic exchange (*duality*) of goods comprises *increment* and *decrement events*, where the former represent events that add value to the company (e.g.receiving cash in a sales activity) and the latter are events that decrease the value of a company (e.g.handing over a product in a sales activity). Each of the events is related to a certain *economic resource* (cash and product in our example, respectively) and to two *economic agents*, the provider and the receiver of the economic resource.

The REA business model language has been used to model various application scenarios, however the data structure of these scenarios has been tailored to the specific needs of that scenario—it was not defined in a domain agnostic way. The approach found in the literature to implement REA applications is to declare concrete instances of entities of the REA meta model as classes in a programming language accompanied by corresponding database tables, see e.g. (Hrubỳ et al., 2006) for a "Pizza Delivery" and (Mayrhofer, 2012) for a "Fish Sale" sample application. In our approach, we are preparing a uniform infrastructure as a generic data model for the implementation of various application scenarios on top. We are thus designing the REA meta model as a multi-layered class hierarchy with corresponding database design and in turn enable specific applications to be modeled and executed based thereon.

Our effort on a generic data model for REA is in its beginnings—first concepts, findings and results have been published in (Gürth, 2014), (Mayrhofer et al., 2014) and (Wally et al., 2014). In this work we present details on the meta modeling methodology we are applying and its implementation concepts.

2 RELATED WORK

(Hrubỳ et al., 2006) is an extensive work on basic and advanced REA concepts and their realization in terms of software models. The presented workflow in the concrete examples follows an intuitive pattern: (i) the REA Meta model is modeled in terms of classes in an object oriented programming language, (ii) the domain specific model is again modeled and implemented in terms of software classes, and (iii) the domain model classes are instantiated at runtime for capturing and representing business artefacts.

In (Nakamura and Johnson, 1998) the Type Object pattern (Johnson and Woolf, 1997) is introduced to describe REA models—this notation is picked up also in (Hrubý et al., 2006; Geerts and McCarthy, 2006): REA concepts are described on a type and on an object layer, where instances on type layer define common properties for instances on the object layer. Also, this *typification* is used as a business model notation: in case a specific REA entity cannot be noted explicitly, its type object is used as a specification for what kind of entity would take part in a certain business pattern.

A model driven approach for the notation of REA based business models is presented in (Gailly and Poels, 2010), where the REA Meta model is defined as an UML profile, and the concrete domain model is modeled as a class diagram applying the REA UML profile. Again, this approach freezes the domain model at design time (expressed in UML) and requires different technology for the definition and the execution of business models.

In contrast to these approaches, we are proposing an integrated solution , where (i) the REA Meta model serves as language definition on MOF[1] layer M2 (modeled and implemented as concrete classes in an object oriented programming language), (ii) the domain model is dynamically modeled in M1 in form of instances of the previously mentioned classes, and (iii) the business artefacts on M0 are again instances of the model instances in M1. This approach allows the user of an enterprise resource planning system that is based upon REA concepts to declare additional exchange methods, resources or agents on-the-fly, and make use of them immediately. To foster M1–M0 interoperability, we strive for an approach that incorporates M1 and M0 models in a single storage and execution system. Interoperability with the approach described in (Gailly and Poels, 2010) could be realized via an import/export mechanism.

2.1 Meta Modeling

In (Atkinson and Kühne, 2008) it is argued that two level modeling is introducing "accidental complexity" in cases where the domain scenario features three or more levels, because some workaround must be found to fit the many levels into the two levels provided. One such application scenario is when an element of a domain scenario requires to influence instances traditionally "out of reach" of that element (declare properties for instances of instances). For that purpose the concepts of "clabjects" (introduced in (Atkinson and Kühne, 2000)) and "deep instantiation/characterization" with "potencies" (Atkinson and Kühne, 2001; Kühne and Steimann, 2004) are introduced. It allows specifying properties that should be instantiated `potency` number of levels below its declaration. However, in our modeling approach, and for the given use case of REA business models, we agree with (Frank, 2011a), in that "potencies $>$ 2 are not needed". In order to be able to model potencies of value 2, (Frank, 2011a) and (Frank, 2011b) introduce the concept of "intrinsic features/attributes/properties" (properties that are not implemented by instances but by instances of instances)—in a sense, intrinsic features are thus deeply instantiated properties with a hardcoded potency of 2. In (Yoder and Johnson, 2002) the "adaptive object-model" architectural style is described by discussing various concepts found in the literature and in implementations for the purpose of providing meta modeling support for the design and implementation of adaptive software systems. As a foundation for such systems the type object pattern and property pattern are identified, which together form a "type square".

In object oriented languages such as Java and C++, a single class declaration/definition is a sketch for four different data related concepts (purposely leaving out behavioral concepts like static or instance methods): (i) it declares the contract of static properties (top-right in Figure 2), (ii) it declares the contract of instance properties (top-left), (iii) it defines values for the static properties (bottom-right), and (iv) instances of that class define values for instance properties (bottom-left). We are currently not explicitly considering methods in our model because (i) in the first step we have no need for adaptive behavior and (ii) we believe that behavior can be well integrated in the approach we are presenting here. In our approach, we are decomposing the three-layered type object pattern into a 2x2 matrix which more closely resembles how class and instance properties are declared and defined in traditional object oriented programming.

3 MODELING APPROACH

In this work we present a structural view on our runtime configurable approach and show how the structural contract of a business model can be modified at runtime without the need for recompilation and redeployment. This structural contract has direct impact

[1] Meta Object Facility (Object Management Group, Inc., 2013), see http://www.omg.org/spec/MOF/

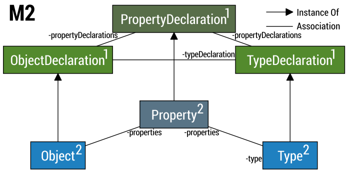


Figure 2: Decomposition of the type object pattern into four elements that correspond to property declaration and instantiation in object oriented programming. Potencies of the types indicate that the top elements are instantiated on M1 and the bottom elements on M0.

on the execution of the business model (mainly by constraining the association of entities and thus reducing the possibilities of configuration).

Specific business models within the REA ontology are traditionally declared in a graphical language based on UML class or object diagram notation[2], which we adopt in the context of this document. They define (i) a taxonomy of entities relevant for that business model (the domain vocabulary) and (ii) the relations between those entities (the business model), e.g. what other entities (mainly agents and resources) certain events interact with or how events relate to commitments.

The remainder of this section will first introduce the concepts we use for the declaration of domain vocabulary and business models (most notable *fragments* and *REA declarations*) and then explain how the declaration is realized.

3.1 Attributes and Associations

`Attributes` are our modeling vehicle to declare variables of built-in data types such as `INT`, `DECIMAL`, and `TEXT`; `Associations` declare complex variables as associations to other `Declarations`. Figure 3 depicts a simplified class diagram of the meta model of the declaration layer M1 in our model. It shows how the REA modeling language is embedded into our model: all REA entities (green: Resource, Event, Agent, ...) inherit from `PropertiedDeclaration`, which essentially means that on the declaration layer, REA entities can be equipped with properties of any kind. Red items (Entity, Declaration, Group, PropertiedDeclaration) represent generic concepts, most notable grouping, cf. (Hrubỳ et al., 2006). Brown items (Property, Attribute, Association) depict attributes and associations, while blue items (Enumeration, Unit, Fragment) represent additional concepts that will be explained below in more detail.

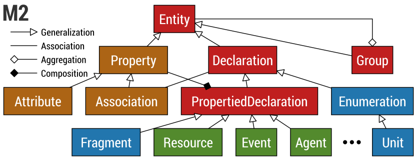


Figure 3: General (simplified) class diagram for M2. Distinct concepts are depicted with different fill colors (explicitly listed and explained in the text). In fact, each of the REA elements (Resource, Event, Agent, ...) is implemented 4-fold, as described in Figure 2, while `Property`, `Enumeration` and `Fragment` are only implemented 2-fold (no typification required).

3.2 Fragments

`Fragments` represent loosely coupled data capsules that, just like core REA entities, inherit directly from `PropertiedDeclaration`, i.e. they can declare named properties (cf. Figures 3 and 4). The purpose of fragments is their reuse in various domain vocabulary declarations or within a single domain. Examples for simple fragments are `HumanName` or `Address`—they have in common that they define rather general concepts that can be used multiple times in the domain vocabulary. In that respect, fragments are very similar to aspects, as described in (Hruby et al., 2006). While the concept of aspects described there is very powerful and even allows the specification of behavior, it is not as smoothly embedded into the modeling layer as our approach, as in our work fragments are first class citizens of the modeling layer and are a standard way of modeling the business vocabulary. In addition, no aspect oriented programming framework is required for usage, and modeling with fragments "feels" just like modeling core REA. Specification of behavior however is one of our elements in the design of a comprehensive software architecture based on the presented data model.

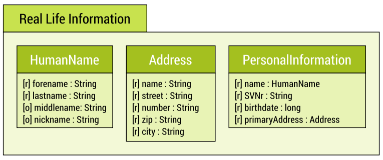


Figure 4: Possible fragment declarations (M1).

In Figure 4 some generic concepts have been modeled as fragments: `HumanName` and `Address` only specify attributes, while `PersonalInformation` is composed of a `HumanName` and an `Address` *association* amongst two other attributes.

[2] While this makes sense, especially from a software engineering point of view, users of business applications are often non-technical domain experts. For that account graphical concrete syntaxes for domain specific languages have been developed (Mayrhofer, 2012; Al-Jallad, 2012).

3.3 Enumerations, Relations and Units

`Enumerations` inherit from `Declaration`, i.e. they can be associated via associations from fragments or REA entities. Enumerations declare a set of strings which are usually related to each other. Examples are fashion sizes (`XS`, `S`, `M`, `L`, `XL`, etc.) or colors (`red`, `green`, `blue`). On the M1 layer, the enumeration is declared only (i.e. its name is defined), however the values for the enumerations are defined on the M0 layer (cf. Figure 5).

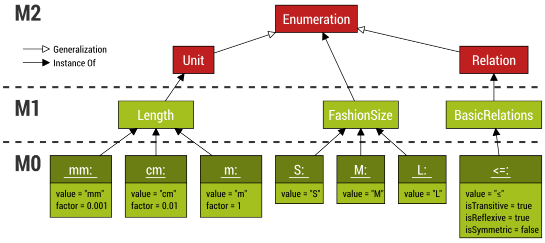


Figure 5: Examples for enumerations, units, and relations.

Units are derived from enumerations and extend them in that they define an additional corresponding factor that relates the values of the enumeration to each other. In the metric system, we can specify length in m, cm, km, etc., in the imperial and US customary systems of measurement length is specified in terms of inch, foot, yard, etc. For the definition of a single unit we therefore need the term needed to describe the unit and the factor that relates the term to the standard unit of the dimension that unit is defined in. Now, at runtime it doesnt matter in which unit a certain value is given, as it can be converted to any other (known) unit easily.

Just like units basically relate different enumeration values by a factor (and thus allow precise conversion from one unit into another), other relations could be defined, too. For instance, the fashion sizes could be related by their size using a "greater than" relation. In fact, the relational information can be modeled completely self-contained, because the relational mappings are defined by a ternary association between enumeration instances: one enumeration instance resembles, the left hand side of the relation, another one the right hand side and a third enumeration instance depicts the operator to be used for the relation. This last enumeration is an instance of the enumeration subclass `Relation` and defines three additional fields: `isTransitive`, `isReflexive`, and `isSymmetric`. Since the operator still is an enumeration instance, its value is of type String—thus the value of the relational enumeration instance could be as simple as "≤", or any literal description.

3.4 REA Declarations

In comparison to fragments that are used to describe domain independent concepts, REA declarations are much more specific, as they implement REA concepts (but still in a domain independent manner when it comes to specific business domains). REA declarations provide the infrastructure to enable (i) the flexible modeling of a specific business domain vocabulary (cf. Section 3.4.1) and (ii) the modeling of the business model (cf. Section 3.4.2).

3.4.1 Domain Vocabulary Declaration

At runtime, REA declarations enable the representation of a business domain in terms of the REA ontology, i.e. instances at the M1 layer are created based on the M2 Meta model infrastructure. These instances on the M1 layer are in turn prescripts for instances on the M0 layer, both of which are managed at runtime. REA declarations are implemented as a set of trees, where each trees root element is one of the core REA concepts, i.e. one tree declares resources, another one declares agents, etc. (cf. Figure 6). In our concept, these declarations define hierarchically structured data capsules only, i.e. each declaration on layer M1 is basically a set of properties (cf. Section 3.1). Behavior is currently not modeled, but support for runtime configurable declaration/definition of methods is planned for the software architecture built upon this data model and thus for future versions of this model.

Each REA declaration is defined by its name, which implies that each name can be given only once. For instance in Figure 6 the given names "Clerk", "Customer", "Sale", etc. cannot be used by any other REA declaration. With this restriction we suppress ambiguities and help designing a cleaner domain vocabulary.

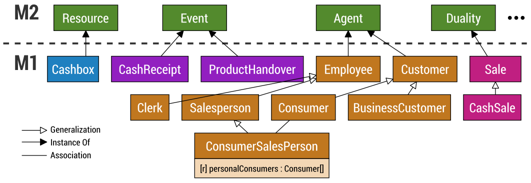


Figure 6: REA declarations: each REA concept spans its own tree of domain vocabulary declarations on layer M1. For the sake of simplicity we exclude further specialization of REA entities, such as the `Event` subclasses `ExchangeEvent` and `ConversionEvent` and their subclasses from this view.

The behavior of a declaration is currently defined by its Meta class only, for instance any kind of agent that is declared on the M1 layer can participate in events, and only the language rules of the REA ontology define the application flow. The provided view on the M1 layer in Figure 6 is a shortcut for what is happening behind the scenes. Based on the power type concept, each entity declaration modeled on layer M1 is in the background split in two entities: an instance declaration and a type declaration—the generated data structure is depicted in Figure 7. The instance declaration holds properties declared for each instance of the given entity, whereas the type declaration holds the class properties that are to be defined only once on the M0 layer in a type instance and are referenced from each of the M0 instances through a `type` association. It is declared in the declaration layer whether the properties of fragments and REA declarations are required or optional—thus the runtime engine needs to check object and type instances for the fulfillment of this domain vocabulary peculiarities.

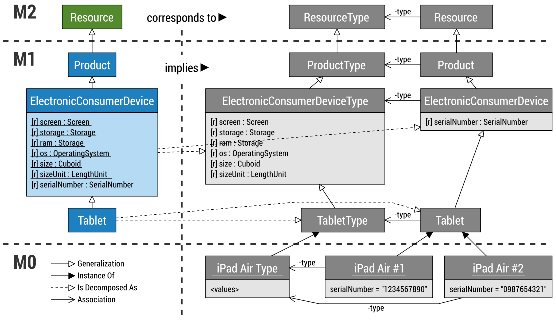


Figure 7: When modeling the domain vocabulary, "behind the scenes" the data structure depicted in grey is generated. The presented properties are all associations, i.e. they either refer to a fragment declaration (this is intended here) or another REA declaration.

3.4.2 Business Model Declaration

Business models are designed at the declaration layer (M1), just like declarations and fragments. For the specification of a value chain of the operational level, the REA declarations need to be arranged accordingly—cf. Figure 8 for a simple sales example, trading products for cash. In this constellation a single sales duality comprises the selling of at least one product (as in a shopping cart) for a single cash payment. At runtime it is rather trivial to check whether an event might occur in a specific duality or not, but of course the inheritance tree must be considered, i.e. instances of any sub-declaration of a REA declaration specified in a constellation are valid entities.

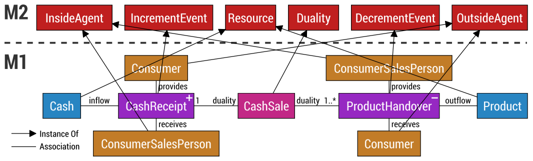


Figure 8: Declaration of a simple "Sales" duality on M1—the cardinality statements on the duality relations specify how often the corresponding event must occur in order to resemble a "valid" duality at runtime (at least one Product must be handed out and only exactly one Cash payment is allowed).

4 CONCLUSIONS

We have presented a data model for the declaration of REA based business models. The concepts of fragments and REA declarations have been sketched and their implementation has been discussed. The presented approach provides a solid basis for the execution of such defined business models in the sense that all required data is available in a single database which can be manipulated and queried at runtime using traditional database interaction methods.

With the separation of declarations and types we have improved the expressiveness of REA models and we have shown that declarations are naturally located on layer M1 while types (of the type object pattern) are really located at layer M0. Our approach provides a consistent view on entities of both layers, and it is thought to be implemented against a single data store, i.e. instances of layers M0 and M1 are "stored together", thus enabling use of referential constraints and referential integrity checking.

We have defined our REA core library with a relational database in the background. Our model provides a class infrastructure for objects and types of the declaration layer M1 and of the runtime layer M0.

We have shown how REA declaration entities can be interconnected in order to resemble business models of varying complexity. With the notion of "fragments", we have introduced a fully integrated new first class citizen in the REA business modeling world. One that does not relate to specific REA concepts but can be applied to any (or none) of them.

One aspect which has not been discussed here is the evolvability of business models, i.e. support for runtime changes in the declaration layer, which is scheduled for further investigation. Also, we are in the process of integrating runtime configurable behavior in addition to the flexible data capsules presented here.

ACKNOWLEDGEMENTS

This work was supported as part of the BRIDGE program of the Austrian Research Promotion Agency

(FFG) under grant number 841287—a joint research effort of Vienna University of Technology and eventus Marketingservice GmbH.

REFERENCES

Al-Jallad, M. M. (2012). REA business modeling language: Toward a REA based domain specific visual language. Student thesis, KTH Royal Institute of Technology.

Atkinson, C. and Kühne, T. (2000). Meta-level independent modelling. In *International Workshop on Model Engineering at 14th European Conference on Object-Oriented Programming*, pages 12–16.

Atkinson, C. and Kühne, T. (2001). The essence of multilevel metamodeling. In Goos, G., Hartmanis, J., and Leeuwen, J. v., editors, *UML 2001—The Unified Modeling Language. Modeling Languages, Concepts, and Tools*, Lecture Notes in Computer Science, pages 19–33. Springer.

Atkinson, C. and Kühne, T. (2008). Reducing accidental complexity in domain models. *Software & Systems Modeling*, 7(3):345–359.

Frank, U. (2011a). The MEMO meta modelling language (MML) and language architecture. ICB-Research Report 43, Institute for Computer Science and Business Information Systems, University Duisburg-Essen.

Frank, U. (2011b). Some guidelines for the conception of domain-specific modelling languages. In Nüttgens, M., Thomas, O., and Weber, B., editors, *4th International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA 2011)*, volume P-190 of *Lecture Notes in Informatics*, pages 93–106, Bonn. Gesellschaft für Informatik, Köllen Druck+Verlag GmbH.

Gailly, F. and Poels, G. (2007). Towards ontology-driven information systems: Redesign and formalization of the REA ontology. In Abramowicz, W., editor, *Business Information Systems*, volume 4439 of *Lecture Notes in Computer Science*, pages 245–259. Springer Berlin Heidelberg.

Gailly, F. and Poels, G. (2010). Conceptual modeling using domain ontologies. improving the domain-specific quality of conceptual schemas. In *10th Workshop on Domain-Specific Modeling*, pages 18:1–18:6. ACM.

Geerts, G. L. and McCarthy, W. E. (1997). Modeling business enterprises as value-added process hierarchies with resource-event-agent object templates. In Sutherland, J., Casanave, C., Miller, J., Patel, P., and Hollowell, G., editors, *Business Object Design and Implementation*, pages 94–113. Springer London.

Geerts, G. L. and McCarthy, W. E. (2000). The ontological foundation of REA enterprise information systems. In *Annual Meeting of the American Accounting Association, Philadelphia, PA*, volume 362, pages 127–150.

Geerts, G. L. and McCarthy, W. E. (2006). Policy-level specifications in REA enterprise information systems. *Journal of Information Systems*, 20(2):37–63.

Gürth, T. (2014). Business model driven ERP customization. Master's thesis, Faculty of Informatics, Vienna University of Technology.

Hrubỳ, P., Kiehn, J., and Scheller, C. V. (2006). *Model-Driven Design using Business Patterns*. Springer.

Johnson, R. and Woolf, B. (1997). Type object. In Martin, R. C., Riehle, D., and Buschmann, F., editors, *Pattern Languages of Program Design 3*, chapter Type Object, pages 47–65. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Kühne, T. and Steimann, F. (2004). Tiefe charakterisierung. In Rumpe, B. and Hesse, W., editors, *Modellierung 2004*, volume P-45 of *Lecture Notes in Informatics*, pages 109–120, Bonn. Gesellschaft für Informatik, Köllen Druck+Verlag GmbH.

Mayrhofer, D. (2012). *REA-DSL: Business Model Driven Data Engineering*. PhD dissertation, Vienna University of Technology.

Mayrhofer, D., Mazak, A., Wally, B., Huemer, C., and Regatschnig, P. (2014). REAlist: Towards a business model adapting multi-tenant ERP system in the cloud. In *8th International Workshop on Value Modeling and Business Ontology (VMBO 2014)*.

McCarthy, W. E. (1982). The REA accounting model: A generalized framework for accounting systems in a shared data environment. *The Accounting Review*, 57(3):554–578.

Nakamura, H. and Johnson, R. E. (1998). Adaptive framework for the REA accounting model. In *OOPSLA'98 Workshop on Business Object Design and Implementation IV*.

Object Management Group, Inc. (2013). *OMG Meta Object Facility (MOF) Core Specification*. Object Management Group, Inc.

Wally, B., Mazak, A., Mayrhofer, D., and Huemer, C. (2014). A generic REA software architecture based on fragments and declarations. In *8th International Workshop on Value Modeling and Business Ontology (VMBO 2014)*.

Yoder, J. W. and Johnson, R. (2002). The adaptive object-model architectural style. In *Software Architecture*, pages 3–27. Springer.

Towards Simulation of Business Processes
Transforming BPMN Models to Enterprise Dynamics Models

Ralf Schepers, Tobias Minning, Yannik Moog and Ingo J. Timm
Business Informatics I, University of Trier, Trier, Germany
{*ralf.schepers, s4tominn, s4yamoog, ingo.timm*}*@uni-trier.de*

Keywords: Business Process Simulation, Material Flow Simulation, Enterprise Dynamics (ED), BPMN.

Abstract: Due to the ISO 9001 certification and process oriented-organization, many business process models are available at enterprises and public institutions. They are e.g. used for documentation or for introducing processes to new employees. As a de-facto standard notation BPMN "Business Process Model and Notation" is widely used. These process models can support static analyses of business processes. Dynamic analysis, e.g., by simulation is beneficial for in-depth analysis and optimization. However, only few approaches are available to perform simulation on basis of BPMN. In the production engineering domain, process simulation for analysis and optimization is a de-facto standard. In this domain, material flow simulation is a valid method of analysis, planning and construction.

In this paper, we discuss the potentials and shortcomings of transforming BPMN models to a material flow simulation model. On the basis of an analysis of BPMN and material flow simulation, we identified requirements for transforming. Four levels of transformation complexity are defined. Furthermore we developed matching relation from BPMN elements to material flow elements. As proof of concept, we implemented the transformation process using Enterprise Dynamics and evaluated its outcome. The benefits and limitations of this approach are discussed in the paper in front of evaluation and related work.

1 INTRODUCTION

The ISO 9001 certification represents an authoritative reference of quality management. Since 2009, Switzerland's public institutions have to be complied with ISO 9001(eCH eGov, 2013). One requirement of this certification is the documentation of business processes. BPMN 2.0 (Business Process Model and Notation) is a common standard for process notation(Freund and Ruecker, 2012), which has been chosen by Switzerland. In consequence, for each certified institution (in Switzerland), the business processes are available in BPMN. Those are used, e.g., for documentation and analysis matters. The modeling and maintenance of business process documents is resource intensive. So an additional value should be added. On basis of the process models, static analysis are enabled, however, as underlying research question of this paper, the question occurs, if it is possible to derive a standard process for dynamic analysis of business processes.

There exists already the approach to simulate business processes by business process simulation. But it lacks on a low leeway in decision-making (Shannon, 1998) and missing standards(Januszczak and Hook, 2011). To re-use existing models in simulation, a transformation approach into executable models is necessary. To improve support of decision-making, the method of material flow simulation will be used, because simulation is a well-defined process method for analysis in engineering. Due to the relevance of BPMN standards and the possibility of the conversion within the standards, the transformation from BPMN models to executable simulation models will be analyzed. As a first step, our objective is to study the transfer of BPMN into an existing simulation environment. This will be evaluated by the implementation of the transformation function in a simulation system i.e. Incontrol Enterprise Dynamics.

2 FUNDAMENTALS

Due to our research objective, BPMN and the mean of material flow simulation will be introduced in the following sections.

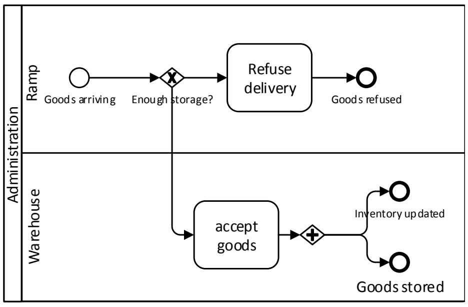


Figure 1: Example BPMN process.

2.1 BPMN

The BPMN 2.0 standard is defined by the OMG Object Management Group(Chinosi and Trombetta, 2012). It includes seven groups of elements. There are activities, gateways, events, swim lanes, data, choreographies and conversation(Chinosi and Trombetta, 2012). Thus, the BPMN elements can be aggregated to a sequence flow. An additional layer can be constructed by message links between the elements. An example of business process can be found in figure 1 and will be used as illustration within this paper.

Although syntax of BPMN is clearly defined, semantics is not. The underlying logic can be annotated or it can implicit. However, the logic have to be extractable. In our example (figure 1) the gateway is annotated with ”enough storage”. Alternatively, it could be annotated with ”insufficient storage”. Also the definition is implicit, which storage is meant. Since 2008 the OMG tries to solve this problem by the definition of ”Semantics of Business Vocabulary and Business Rules” (SBVR)(OMG, 2008). This provides the ability to map the context to a machine-interpretable operands. It is also used to derive processes from textual descriptions(Bajwa et al., 2011). However the problem of implicitly can only be solved by improving the modeling process.

Beside the semantics, the general transformation and simulation potential of BPMN must be analyzed. For this purpose, the technical basis of transformation will be considered. In order to recondition BPMN diagram information in a machine-interpretable way, the OMG defines the ”BPMN Diagram Interchange” schema. BPMN DI is part of the BPMN 2.0 standard. The DI scheme includes all structural information of the model. Meta-information, such as the element type, its name or edge type, and the execution semantics are included in the ”BPMN Execution Semantics Schema” (BPMN-ES)(OMG, 2011). A BPMN model consists of the Diagram Interchange and the Execution Semantics Information. It can be stored by a XML file. This forms a reasonable technical basis, which includes the information needed to transfer and display BPMN processes between different modeling platforms.

2.2 Material Flow Simulation

In a simulation, a model of a real world system, will be run under experimental conditions (Shannon, 1998). The objective is to understand the behavior of the system to find improvements, which can be evaluated through experiments (Shannon, 1998). A simulation project consists of analysis, modeling, experiment and interpretation of results(Lattner et al., 2011). Each simulation run is performed several time due to its probabilistic parameters (Lattner et al., 2011). Material flow simulation is one kind of simulation with focus on production and logistic. An abstract view on material flow simulation is given by (Rittgen, 1998). A process consists of elements which are atomic in nature. The process is define by its elements. Each element can perform an action, which is hidden, only the effect of a process can be seen. But this view is too abstract to be used as a transformation basis.

The mean of material flow simulation is based on experience and efforts of engineers, who are the main user of those systems. Different commercial simulation software, e.g. “Plant Simulation” by Siemens[1], “Enterprise Dynamics” (ED) by Incontrol Simulation Solution[2], and “Arena” by Rockwell Automation[3] are available. There notations are based on graph models, however different elements are obtainable. As an example of different elements, Arena offers a “decide” module comparable to a BPMN gateway. This function has to be implemented in ED and in Plant Simulation due to elements properties or adapted elements. As a representative of a material flow simulation software, we will further use ED. It will be exemplarily introduced.

ED offers an atomic library of elements. Each element is represented by an instance of an atom, with its own variables and logic. So, ED combines both an object orientation and an event implementation. As a result, the atoms can be defined, reused and derived (object-orientation). The logic, however, is implemented inside the instance of an atom (event-based). The tokens (e.g. products or orders) are created by a source and leave the process via a sink. The status of the atoms and global variables, like the set of tokens within the model or the content of a queue, are

[1] www.plant-simulation.de

[2] www.incontrolsim.com

[3] www.arenasimulation.com

representing the state of the model. After creation, the tokens are then processed according to the model pattern.

2.3 Discussion

From a methodological point of view, business processes are similar to material flow models witch respect to its graph structure as well as its atomic elements. However its perspective can be different. Business processes, which are modelled for documentation matters, should map all relevant elements of its business objectives. In contrast a material flow model, is a problem-oriented homomorphous mapping of a real world system. This could be congruent to business processes, but it is not compulsorily. In case of the usage of an existing BPMN model, the focus of the model possibly have to changed in order to meet the underlying question of the analysis.

3 REQUIREMENTS ON SIMULATION SOFTWARE

The simulation environment must be defined, before raises the possibility of a transformation, the overall requirements to.

1. Availability of basic concepts of nodes and edges
2. Decision functions within nodes
3. Model import via XML format
4. Graphical representation of BPMN model
5. Interface for input/ output data
6. Validated random generator

Starting with the sixth requirement, a validated random generator is necessary to receive statistically valid results(L'Ecuyer, 1997). Random values are used as an abstraction of variance of the real world to a model. A validated random generator should be included in a commercial and established simulation software. The basis of an analysis in simulation lies in the analysis of data. However, it is not a typical job to a business process modeler. In contrast, data handling, is a core task of the method of material flow simulation. Well studied solutions for data analysis are available, e.g. by (Bogon et al., 2012).

In the case of transforming existing models into a material flow simulation software, a graphical representation, close to BPMN would reduce training. The graphical representation leads to an increased acceptance of the method of simulation, because the participants understand the model. This is a crucial point in simulation projects(Wenzel et al., 2007). To perform simulation experiments, the gateways have to be negotiable. So there is a need of decision functions inside the available nodes. Close related to the graphical representation, is the ability of the model import. This is not yet possible to the above mentioned software solutions.

The first four requirements could be covered by common BPMN simulators such as Signavio[4]. However, the last requirements, necessitate the use of more extensive simulation environments, such as material flow simulation software solutions. The use of material flow simulations demands a high variance in modeling, due to corresponding production. Therefore it provides expandability of the used elements. Manufacturing processes models follows similar to BPMN models, the pattern of change between puffer, processing and transportation.

4 CONCEPT

At a high level of abstraction, the generation of a transfer process of business process to material flow simulation model, can be seen in figure 2. At the beginning the business model has to be readable, e.g. via a XML file. Subsequently transfer rules from a meta-model of business process or specific a meta model (e.g. BPMN meta-model) to a meta-model of material flow simulation have to be defined[5]. Afterwards, the rules can be deployed on the business process model. As a proof of concept, whether a transformation is possible, direct transformation rules will be specified in the following. The abstraction steps will be avoided.

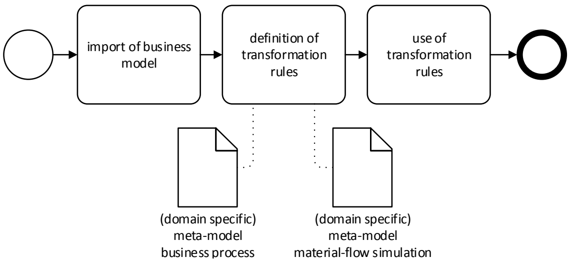

import of business model
definition of transformation rules
use of transformation rules
(domain specific) meta-model business process
(domain specific) meta-model material-flow simulation

Figure 2: Generation of a transformation process.

4.1 Matching Concept

As mentioned in section 2.2, there are differences be-

[4] www.signavio.com

[5] So far no meta-model of material flow simulation is known to the authors.

Table 1: ED atom overview.

Atom	Function	Input / Output	Attribute
source	source of product	product / n	inter-arrival time
sink	leaving of product	1 / -	-
server	cycle time	n / n	cycle time,in- & output strategy
splitter	sequence flow split	1 / n	cycle time & quantity matrix
assembler	sequence flow merge	n / 1	cycle time & quantity matrix
composition container	element group	- / -	-

tween available simulation solutions, and there exists no consensus of notation. To prove the transferability, a direct transformation to ED will be discussed. In the following, destination elements in ED will be searched, to map BPMN elements. Thus standard properties and behaviors according to the BPMN standards could be defined in advance. Alternatively, the properties of existing ED elements can be used.

To model the production process flow-logic and the presentation of the processing activities, ED provides six basic elements (atoms): source, sink, server, assembler, transport, and composition container. The products are distributed via directed channels, connected by in- and output channels. An overview of atoms, their functions, and their properties can be seen in table 1. The processing is provided by the server, the assembler and the splitting atoms. Here, the choice of the output channel (send to), as well as the processing time can be fixed by stochastic distribution or by an absolute term. An example of processing atoms, are packing and unpacking activities. The split atoms or the assembler atoms are able to split or to join the process flow. A splitter atom can be entered via one input channel and *n* output channels. Vice versa to the assembler atom. The determined in- and output ratio of products can be set by a table. Also a processing time can be set.

For visual grouping of the atoms, ED uses a composition container atom. It could be also used for a logical grouping. The atom provides a rectangle that is drawn around atoms. According to the association of data objects, ED uses an ActiveX or a database connection. Thus, the simulation can be based for example on an Excel spreadsheet, containing input data. Of course data elements in the BPMN perspective, like standard forms are not needed for simulation.

The prescribed ED atomic structure potentially offers both the modeling options to map the BPMN flow elements, as well as the possibility to represent the process. A mapping of BPMN to ED elements can be seen in table 2. On this basis a transformation possibility can be derived to infer the suitability of ED as a target simulation environment. In addition, a logic detection is needed. In the following, four different levels of complexity of transformation can be differentiated.

4.2 1st Degree: Straight Mapping

The straight mapping includes presentation of the basic elements, shown in table 2. A 1:1 transfer of elements to ED takes place. Simple tasks can be mapped to server atom with appropriate properties. From the group of gateways, the parallel and the exclusive gateways can be transferred. The parallel gateway can be matched to the splitter atom. The process flow can be merged by an assembler atom. In case of an exclusive gateway, a server atom with specific “sent to” ratio will be set. This can be set by conditions, as well as a percent ratio. The merging gateway can be set to a server atom. Such takes the product from any input channel and supplies it to one output channel. Informations about average processing time of tasks is not given by a BPMN model. Consequently it would have to be inserted while the transformation process. To get plausible simulation results, data have to be collected. By default, this follows a negative

Table 2: Mapping complexity degrees.

BPMN	Enterprise Dynamics
1^{st} degree	
start-event	source
end-event	sink
timer-events	time based release
parallel gateway	splitter/ assembler
exclusive gateway	output strategy server
sequence flow	relations
task	server
(swim) lane	composition atom
2^{nd} degree	
start message event	adapted source
message event	adapted atom
message task	adapted token
sub process	composition atom
3^{rd} degree	
looping task	adapted server
event based gateway	adapted server

exponential function, set by ED.

4.3 2nd Degree: Local Logic Detection

The 2^{nd} complexity degree requires to implement the rules based on the local recognition of logic. It is based on the defined elements to represent the atoms. In this case, these rules primarily support message events. This requires the adaption of ED atoms, because communication actions are not a part of the material flow simulation method yet. There are possibility to send messages like parameters, but a deliberation is not yet possible. We did some work, to implement communication actions according to FIPA standard(Bellifemine et al., 1999).

4.4 3rd Degree: Global Recognition

In order to map more complex control structures like anonymous processes and pools, a composition of ED atoms is required. Also sub processes and black boxes can be represented. For this purpose, incoming and outgoing messages as well as sequence flows of containers have to be identified and grouped. They could lead to a source or a sink atom in a composition container.

For this level of complexity, pattern recognition is necessary. It should check the kind of parent and son element. The event- based gateways can be viewed in a global context with their downstream events. The event gateway can be mapped to a server with its own processing strategy. This can be influenced by specifying a script in the ED scripting language 4DSCRIPT. The in- and output channels of the group are then aggregated into a server atom. Likewise, the mapping of loop tasks can be classified under this level of complexity.

4.5 4th Degree: SBVR Logic and Complex Mapping

The 4^{th} level includes the mapping of the remaining elements. Although these affect the process flow, the underlying logic implemented is difficult to recognize, and consequently more difficult for a machine due to lack of semantic interpretability. The implementation requires an extended semantic context, such as it is used for example in the SBVR research. To implement the logic correctly, additional recognition of complex patterns as well as dynamic transfer rules will be necessary.

5 EVALUATION

Starting to evaluate this approach, a transformation application based on 1^{st} degree, has been implemented. The transformation fulfills the requirements three, by acting as an interface between a BPMN model and ED. It includes importing of the BPMN XML documents, as well as conversion and storage in the file format used by ED. Due to the application, ED is able to fully implement all transferable elements. As shown in figure 1, the application were able to transform all elements. The atom properties, like cycle time were predefined. In comparison to figure 1, which were used as origin, requirements one (availability of basic concepts of nodes and edges) as well as the "graphical representation of BPMN model" can be proven. Due to the predefined parameters, output data could be generated by ED (requirements five) but there were no further transformation as 1^{st} degree complexity. In addition, no data collections were made to set parameters. Those were set random. As a result, generated data are not valid. For each control structure of the 1^{st} and 2^{nd} complexity degrees a mapping could be found. In addition, a corresponding control structure has been developed to map the 3^{rd} level of complexity. For a transformation of higher degree than 1^{st} degree, a pattern and logic recognition is necessary. However, the author belief, that the information base should be adequate up to 3^{rd} degree, but a composition of atoms is required. The mapping of further elements and control structures of the 4^{th} degree of complexity underlie the mentioned problems. They cannot be transpose without an extension of the information base. The requirement for a correct representation of the process flow logic is given only for the first three levels of complexity.

6 RELATED WORK

An approach to transform BPMN to a simulation environment was proposed by (Cetinkaya et al., 2012). They developed an executable meta-model of the BPMN standard. But they weren't able to establish an executable model. In addition, no reproduce of the diagrams was possible. Another possibility to simulate business processes are proposed by (Mueller, 2012). They are proposing to use EPC (event-driven process chain) notation to model simulation models. Those models were then compiled to an executable program. This forms an advantage in contrast of our approach, because no standard notation or software are used to run the process model. However shortcoming were identified due to EPC extension for simulation like

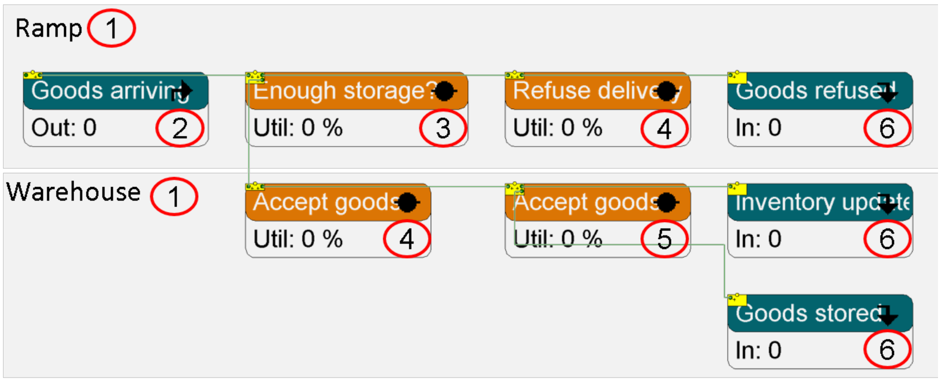


Figure 3: Process in ED: 1 swim lane, 2 start-event, 3 parallel gateway, 4 task, 5 exclusive gateway, 6 end-event.

data. Also simulation experts should be necessary. In case the use of our approach, also simulation specific data are needed. The advantage of our approach lies in the reuse of existing models as well as the use of material flow simulation software. In contrast to generated code, adjustment can be done with respect to modeling and configuration actions. This can be done and understood by BPMN modeler too after training.

(Januszczak and Hook, 2011) discuss a simulation standard for business process managements in general. In this approach, also the usage of a standard notation (e.g. BPMN) is proposed to reduce the needed training for simulation. The discussion remained at a meta-level and proposed no transformation but an extension of business process modeling notations. In consequence, BPMS (business process management suite) can be used for simulation. In our approach, the use of material flow simulation software is proposed, to use existing method(s), as well as existing software solutions.

Another approach is represented by (Dijkman et al., 2008). They map BPMN to petri net. In difference to material flow simulation and BPMN, petri net have got a well-studied formal language as well as “efficient static analysis techniques”(Dijkman et al., 2008). As already mentioned, semantics are identified as a challenge by (Dijkman et al., 2008). A similar approach is presented by (Raedts et al., 2007). A lot of BPMN primitives are successfully mapped into a petri net. The subjacent question was to be able to find specification inconsistency located within the BPMN diagrams like deadlocks or loops.

In summary, the tendency to generate an additional advantage of the models gets obvious. Also the ability to convert between the notation and methods are met. In case of transformation, the problem of machine-interpretability demand of an ontology for business process(Cabral et al., 2009).

7 CONCLUSION

BPMN as a modeling language is well established in business. One benefit of this approach is, to improve the communication between the principal and agent due to a joint basis of communication in a simulation project. Thus, as an additional application of our approach, BPMN models could be specified in workshops between customers and service providers of simulation technology. Doing so, a first scratch, i.e., a first simulation model could be derived from these models. As a missing link, semantics, especially the quantitative information of the models, are required for sophisticated simulation. Here more research is needed to identify the best point of time, when to acquire these information in the simulation process. Depending on this result the acquisition should extend the BPMN or the material flow model.

As there is an extensive usage of BPMN models in companies, adding additional value to these process models is of high relevance. In this paper we analyzed, how BPMN process can be transferred into ED material flow simulation to enable a dynamic analysis of the underlying process structure. The feasibility of transferring a BPMN process model into a simulation model of commercial material flow simulation software has been shown at a defined (low) level. At higher level, the relevant tasks have been discussed. As a result of the analysis, there is a high coverage of concepts in BPMN and in material flow simulation.

We proposed a general transformation process (see fig. 2), by which a transformation will be run through a meta-level. By the employment of BPMN meta-model or the Business Process Definition Meta-Model, a transformation to a meta-model of simulation is needed. In contrast, the presented approach discuss a direct transformation. Based on section 2.2 argumentation, that different simulation software have got different natured, but similar notations, the presented approach is limited to following aspects:

1. static specific rules (BPMN to ED)
2. portability of approach to other material flow simulation is not ensured
3. portability of approach to other business process modeling notations like EPC is not ensured

The negotiability to other material flow simulation software systems as well as other modeling notations is disputable. Difference between modeling methods (e.g. focus) of other software and notations, lead to different used elements. As a result, there is no standardized meta-model of material flow simulation. However, relevant tasks, e.g. logic detection remain relevant to other modeling notations as wells as to other material flow simulation software. Further research steps will include the analysis of common material flow simulation software. In comparison to BPMN notation, an ontology based approach will be developed. Goal is the interoperability of BPMN to material flow simulation software.

REFERENCES

Bajwa, I. S., Lee, M. G., and Bordbar, B. (2011). Sbvr business rules generation from natural language specification. In *AAAI Spring Symposium: AI for Business Agility*. AAAI.

Bellifemine, F., Poggi, A., and Rimassa, G. (1999). *JADE - A FIPA-compliant agent framework*, pages 97–108. The Practical Application Company Ltd.

Bogon, T., Jessen, U., Lattner, A. D., Paraskevopoulos, D., Schmitz, M., Spieckermann, S., Timm, I. J., and Wenzel, S. (2012). Towards assisted input and output data analysis in manufacturing simulation: The EDASim approach. In Laroque, C., Himmelspach, J., Pasupathy, R., Rose, O., and Uhrmacher, A. M., editors, *Winter Simulation Conference (WSC 2012)*.

Cabral, L., Norton, B., and Domingue, J. (2009). The business process modelling ontology. In *Proceedings of the 4th International Workshop on Semantic Business Process Management*, SBPM '09, pages 9–16, New York, NY, USA. ACM.

Cetinkaya, D., Verbraeck, A., and Seck, M. D. (2012). Model transformation from bpmn to devs in the mdd4ms framework. In Wainer, G. A. and Mosterman, P. J., editors, *SpringSim (TMS-DEVS)*, page 28. SCS/ACM.

Chinosi, M. and Trombetta, A. (2012). Bpmn: An introduction to the standard. *Computer Standards & Interfaces*, 34(1):124–134.

Dijkman, R. M., Dumas, M., and Ouyang, C. (2008). Semantics and analysis of business process models in bpmn. *Inf. Softw. Technol.*, 50(12):1281–1294.

eCH eGov (2013). ech-0073 dokumentation.

Freund, J. and Ruecker, B. (2012). *Praxishandbuch BPMN 2.0.* Hanser, Mnchen; Wien, dritte edition.

Januszczak, J. and Hook, G. (2011). Simulation standard for business process management. In *Simulation Conference (WSC), Proceedings of the 2011 Winter*, pages 741–751.

Lattner, A. D., Pitsch, H., Timm, I. J., Spieckermann, S., and Wenzel, S. (2011). Assistsim–towards automation of simulation studies in logistics. *SNE*, page 119.

L'Ecuyer, P. (1997). Uniform random number generators: A review. In *Proceedings of the 29th Conference on Winter Simulation*, WSC '97, pages 127–134, Washington, DC, USA. IEEE Computer Society.

Mueller, C. (2012). Generation of epc based simulation models. In Troitzsch, K. G., Möhring, M., and Lotzmann, U., editors, *ECMS*, pages 301–305. European Council for Modeling and Simulation.

OMG (2008). Semantics of business vocabulary and business rules (sbvr), v1.0. Technical report, Object Management Group.

OMG (2011). Business Process Model and Notation (BPMN), Version 2.0.

Raedts, I., Petkovic, M., Usenko, Y. S., van der Werf, J. M. E. M., Groote, J. F., and Somers, L. J. (2007). Transformation of bpmn models for behaviour analysis. In Augusto, J. C., Barjis, J., and Ultes-Nitsche, U., editors, *MSVVEIS*, pages 126–137. INSTICC PRESS.

Rittgen, P. (1998). *Prozeßtheorie der Ablaufplanung.: Algebraische Modellierung von Prozessen, Ressourcenrestriktionen und Zeit.* Teubner-Reihe Wirtschaftsinformatik. Teubner B.G. GmbH.

Shannon, R. E. (1998). Introduction to the art and science of simulation. In *Proceedings of the 30th Conference on Winter Simulation*, WSC '98, pages 7–14, Los Alamitos, CA, USA. IEEE Computer Society Press.

Wenzel, S., Collisi-Böhmer, S., Weiß, M., Rose, O., and Pitsch, H. (2007). *Qualitätskriterien für die Simulation in Produktion und Logistik: Planung und Durchführung von Simulationsstudien.* (VDI-Buch) Chemische Technik Verfahrenstechnik. Springer.

On Advanced Business Simulations
Converging Operational and Strategic Levels

Marc Drobek[1,2], Wasif Gilani[1], David Redlich[1], Thomas Molka[1] and Danielle Soban[2]
[1]*SAP UK Ltd., Belfast, U.K.*
[2]*Department of Mechanical and Aerospace Engineering, Queens University Belfast, U.K.*
{marc.drobek, wasif.gilani, david.redlich, thomas.molka}@sap.com, d.soban@qub.ac.uk

Keywords: KPI predictions, Business Dynamics, Business Process Simulation, Process Performance Parameters, Business Simulations, Semantic Knowledge, Ontologies.

Abstract: Business Dynamics (BD) enables strategic Key Performance Indicator (KPI) predictions to monitor the health status of companies and support the decision making process. Nevertheless, a very important factor, which is generally overlooked, is that the top level strategic KPIs are highly influenced by the operational level business processes. These two domains are, however, mostly segregated and examined as silos with different solutions. In this paper, we are proposing a framework for advanced business simulations, which converges the two domains by utilising Ontologies and process execution data. Establishing this connection enables drilling down from a high level KPI perspective into the underlying operational level details to discover hidden bottlenecks and pre-emptively apply corrective actions.

1 INTRODUCTION

Managing global companies is an extremely challenging task, which needs a lot of expertise and experience. These companies are highly complex ecosystems, with millions of customers and thousands of employees organised in various departments in different geographical locations. Like every other complex ecosystem, these need to be managed carefully and with huge responsibility to keep them flourish and stimulate growth. KPI monitoring and prediction solutions, based on multiple concepts (for example, database reporting tools, time series analyses or Business Dynamics (BD) (Sterman, 2000)), are generally employed to keep a check on the company's performance, foresee future development and make critical decisions. Examples of conventional strategic KPIs, which are generally monitored, are revenue, profit, number of orders, employee turnover rate, customer satisfaction, etc. These KPIs mainly relate to business objects, (e.g. Sales Order, Customer, Employee, etc.) and in most cases are computed based on the actual data contained in the business objects (e.g. sales orders in case of sales revenue). A very important factor, which is generally overlooked, is that these strategic KPIs are highly influenced by the operational level business processes, which are the foundation pillars of any company, and are orchestrated to offer the services or products that the company deals with. An efficient execution of these processes is therefore vital for company's success. Huge amount of event data (process logs), is generated during process execution, which has only recently received attention by the business world and research community. The performance indicators computed from execution data, called Process Performance Indicators (PPIs), are used to evaluate the performance of business processes (Ann et al., 2011; Del-Rio-Ortega et al., 2010). Such PPIs are, for instance, process queue length, throughput, resource utilisation, instance occurrence, etc. This process execution data, therefore, holds the key to uncover problems and bottlenecks at the business process execution level. Significant research work has been carried out in the area of PPI extractions, for instance, in the context of Process Performance Mining and Business Process Performance Management (Redlich and Gilani, 2011; Fritzsche et al., 2009; Heilig and Möller, 2014).

However, so far KPIs and PPIs have mostly been dealt with and consumed in isolation at different levels (strategic and operational). For example, if one looks into the widely adopted commercial solutions in the Business Intelligence domain, such as Business Objects, Aris WebMethods, Oracle BAM, SAP Process Observer, they all deal exclusively either with

PPIs or KPIs (Howson and Newbould, 2012; Hecking and Schroder, 2013). The authors have not come across any work that connects the PPIs and KPIs, except a commercial solution by Software AG that offers a manual mapping approach to connect these two levels (SoftwareAG, 2014). But clearly, there is a connection between the two levels, as inefficient execution of BPs eventually leads to KPI deviations, which might cause financial collapse of the company. Simple examples highlighting this deep connection between PPIs and KPIs are:

- An inefficient execution of Sales Opportunity Management process in a sales office leads to decreased revenue.
- A delayed Consignment Fill-up process leads to lost sales.
- An imperfect execution of the Idea to Market (I2M) process leads to companies failing to introduce novel competitive solutions, thus loosing market share (e.g. Kodak and Blackberry).

In this paper, we propose a systematic framework for BD simulations which utilises semantic knowledge sources, simulations and PPI analyses, to explore and enhance KPI predictions. This paper is therefore split into the following sections: Section 2 describes the available operational data and the two approaches that are generally used for PPI predictions. Section 3 gives an explanation of KPI predictions with BD simulations and highlights the need of semantic knowledge to generate such predictions. Section 4 introduces our advanced business simulation framework and design decisions that have been made, to incorporate PPI predictions in BD models with the goal of enhanced KPI predictions. We further outline the need for additional semantic knowledge sources, necessary to describe the dependencies between KPI-KPI and KPI-PPI to create KPI predictions. Finally, in Section 5, we conclude the paper and list further research challenges, which need to be tackled in future work.

2 BUSINESS PROCESS ANALYSES

Software systems supporting the execution and management of operational BPs are called Business Process Management Systems or Business Process Management Suites (BPMSs) (Ko et al., 2009). Examples of BPMSs are SAP Netweaver BPM (Woods and Word, 2004) or Intalio BPMS Designer (Intalio, 2013). When BPs are executed they produce events each representing a transition in the system's state.

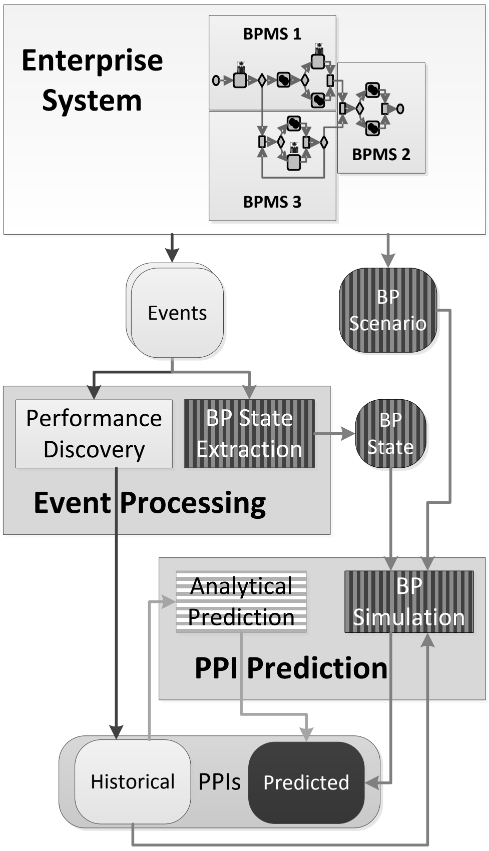


Figure 1: Information flow for PPI Extraction and Prediction: (1) Analytical Prediction (horizontally striped); (2) Prediction via Simulation (vertically striped).

These events are usually of a simple nature and often only comprise raw information, like process instance id, timestamp, and type of the state transition but not the state of the whole system (Van Der Aalst, 2011). One example of such an event is:

```
2013-05-26 T 13:45 CET: Activity "Check
  availability" completed, pi-id:  253
```

The performance of the BPs is measured with PPIs, e.g. *activity net working time* - the elapsed time of an execution of a single activity, or *activity throughput* - number of executions of a single activity per time period, or *process instance occurrence* - how often the process has been initiated. The historical PPIs are computed by capturing, selecting, aggregating, and eventually abstracting raw events from process logs to generate high-level performance information about the system (Performance Discovery).

The prediction of PPIs is generally carried out with two different approaches: The first approach is to utilise existing data-centric Business Intelligence tools to predict each PPI individually based on its history, i.e. *Analytical Prediction*. These numerical and/or statistical methods do, however, not take the workflow information that is readily available in many BPMSs into account (Redlich and Gilani, 2011). The second approach, which includes workflow information to create more meaningful prediction results, is *Prediction via Simulation*. In this second approach, in addition to the extracted historical performance data, *BP Scenario* information about control workflow, involved roles and resources are utilised in a discrete event simulation (Robinson, 1964). The beneficial effect of using simulation over analytical methods for predicting PPIs is discussed in (Redlich and Gilani, 2011) and (Porzucek et al., 2010). Figure 1 shows the general concept for extracting *Historical* (and current) PPI data plus the two approaches of how to compute the *Predicted* PPI data via Analytical Prediction (horizontally striped) and Prediction via Simulation (vertically striped).

3 BUSINESS DYNAMICS

Predicting KPIs in enterprises is a commonly used method to support the decision making process to line up the future business strategy. These predictions are usually carried out with time series analyses of historical KPI data (Brockwell and Davis, 2006). However, in large businesses, KPIs appear to be high-dimensional, non-linear, are part of feedback loops and not isolated. Especially the fact, that KPIs are no silos, thus are being influenced by a variety of other KPIs and variables, raises the level of mathematical expertise needed to perform time series analyses. In such cases, when the system under study is highly non-linear and contains feedback, Forrester's System Dynamics concept is in general well suited (Forrester, 1961). Sterman already showed the applicability of Business Dynamics (BD) in the business domain (Sterman, 2000). Since BD is essentially a specialised SD concept, it adopts the same traditional SD steps to support the modeller in understanding the business and creating predictions. SD itself is, however, already a well established concept and various steps involved in SD have been debated for decades (Burns, 1977; Ford, 1999; Binder et al., 2004). We have summarised these traditional steps in the lifecycle figure 2, using Burns and other sources.

The process usually starts with eliciting knowledge from the business domain experts (company employees, BP owners, managers, directors and so on) and formalise it into *Causal Loop Diagrams*. CLDs capture the most important business variables (revenue, sales, orders, customer satisfaction) and their inter-connections (Burns, 1977). The next phase is the transformation of CLDs into *State/Flow Diagrams* (SFDs), which capture the resources/material flowing through the business (Forrester, 1961). In the next step, the SFDs are annotated with parameters and equations, which embodies a variety of different limitations (Drobek et al., 2013), and then fed into the simulation engine (e.g. Vensim, Stella (Richmond and isee systems (Firm), 2008)) to finally carry out simulations. Once first simulation results have been produced, the modeller has to evaluate, whether the output matches the real-world behaviour. Further iterations are executed to improve simulation results.

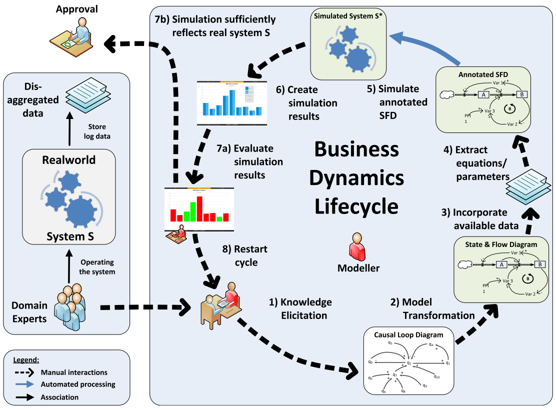


Figure 2: The traditional BD life cycle.

The BD modelling process is, however, not trivial, since it is mostly based on the modellers understanding and knowledge of the target business. For instance, the modeller is expected to:

- semantically link together the KPIs and their influencing variables in the CLD (e.g., profit is influenced by monthly expenses)
- detect and model feedback loops
- determine the resources/material flowing through the system to create SFDs (e.g., money, customers, satisfaction)

To find these connections, the modeller usually relies on the business domain experts (Forrester, 1991). Mostly, this knowledge is a mental model and needs to be manually extracted by the modeller, which gives room for misinterpretation and failure (Ford and Sterman, 1998). Additionally, the modeller has access to the historical business data (KPIs, documentation, reports), which she uses to extract the dependencies and relations of the target KPI. But even with this knowledge, modelling CLDs and SFDs is still a very challenging task, since the identification of the important variables, which influence the target KPI and main feedback loops, requires a lot of experience, expertise and imagination.

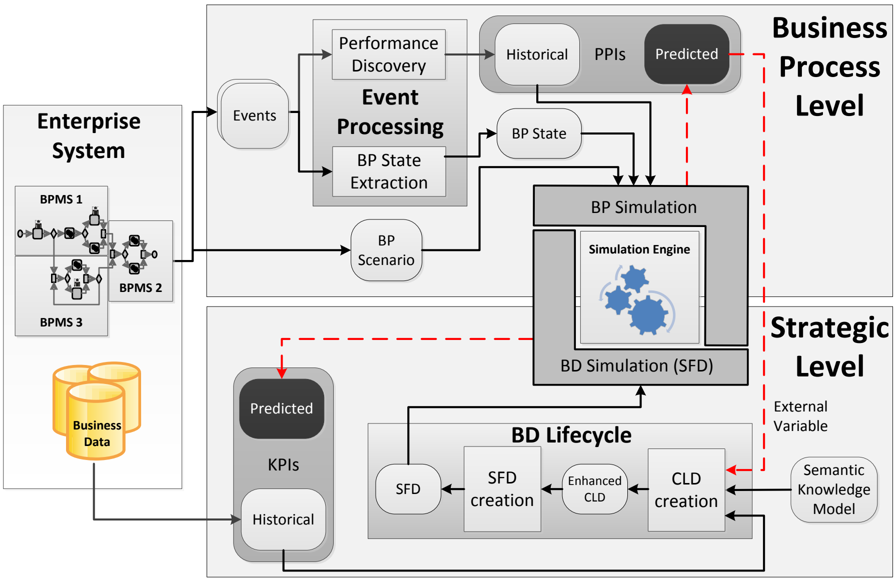


Figure 3: Framework for linking strategic KPIs with operational PPIs.

4 AN ADVANCED BUSINESS SIMULATION FRAMEWORK

As stated earlier in the introductory section, and highlighted with examples, the strategic KPIs are highly influenced by the execution of BPs at the operational level. However, the strategic level decision makers (head of sales, board members, etc.) lack the process level visibility to make informed decisions. Establishing a link between the PPIs and KPIs will enable this process visibility. Our proposed framework solution, in addition to the strategic KPIs, also incorporates the predicted PPIs in the BD life cycle, thus establishing a link between the operational and strategic level. With these links, our framework enables identifying and addressing issues and bottlenecks pro-actively at the operational level before they start impacting the strategic KPIs. Figure 3 shows a schematic description of our advanced business simulation framework. The two main approaches to compute PPI predictions are described in Section 2. Our framework employs the second approach, *Prediction via Simulation*, because it preserves the control flow information of the targeted BPs and thereby helps to exploit the benefits of behavioural simulations (Porzucek et al., 2010). Additionally, the KPI prediction process via BD simulation is orchestrated following the BD life cycle provided in Section 3 (shown in figure 2). The connection between the operational and strategic level is established by including the PPIs in the CLD creation process. This is done by applying causal indicators, such as correlations and Granger causality (Granger, 1969), among the KPIs and PPIs and further extracting their semantic dependencies from available enterprise ontologies, as described later. The precomputed PPIs are an additional input data source, when designing these *enhanced CLDs* in our framework. Once an enhanced CLD is transformed into an SFD, it is simulated and finally KPI predictions are generated. In a standard BD simulation run, each element apart from static parameters or converters is simulated. Since the PPIs are more accurately calculated via event processing and BP simulations (as shown in figure 1), they are not recomputed again in the BD simulation. The reason behind this accuracy is the availability of highly formalised and well structured behavioural models and event data that enables an automated prediction process. The PPIs are therefore, by definition, considered to be parameters or converters in a BD simulation run, even though, they also change over time. Our solution introduces a new BD element type called *”external variable”*, which maps to a precomputed PPI. Such an external variable is not influenced by any of the other BD elements, but is still continuously updated with each simulation run at the operational level.

The creation of CLDs has always been a non-trivial task, as discussed in Section 3. By introducing an additional operational PPI input, this problem becomes even more challenging. This raises the need for a definition of some sort of a *”dependency model”* to provide a guideline on how to link together operational data to the KPIs. Such a model needs to describe the relations and dependencies be-

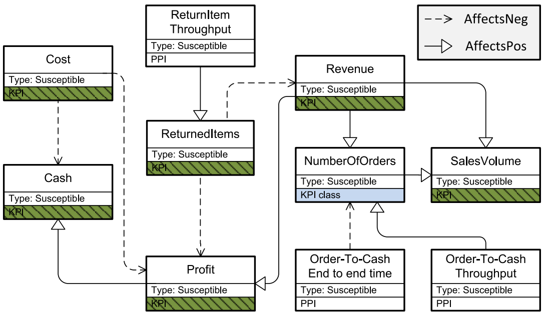


Figure 4: Visual representation of a business ontology to describe the relation between KPIs and PPIs.

tween KPI-KPI and KPI-PPI, thus reflecting the semantic knowledge that usually resides in the mental models of the domain experts. Ontologies and knowledge graphs are two examples for formal ”semantic knowledge models” (SKM) (Zhang, 2002). Companies, such as Google, are using knowledge graphs to *”... understand real-world entities and their relationships to one another”*, which are either automatically harvested from the web or are once manually created by the domain experts and then reused (Steiner et al., 2012). We find Ontologies to be well suited to describe this kind of semantic information and have included such ”semantic knowledge models” into the framework, as shown in figure 3. A snapshot of an example retailer business ontology is shown in figure 4. This picture visualises the connection between some high-level KPIs, for instance, Revenue, NumberOfOrders, ReturnedItems, and three PPIs, namely *ReturnItem_Throughput*, *OrderProcess_Throughput* and *OrderProcess_EndToEndTime*. These three PPIs are reflecting the throughput of the ReturnItem and Order-To-Cash BP, as well as the average execution time for one Order-To-Cash BP instance. Additionally, the KPIs and PPIs are connected via two relations: *affectsPos* and *affectsNeg*. Whilst the affectsPos relation suggests a positive influence from source to target element (e.g. directly proportional), the affectsNeg relation negates this dependence (e.g. inversely proportional). In this given case, we know that the Order-To-Cash BP drives the high-level KPI NumberOfOrders, which then impacts the sales volume and finally the overall revenue of the company. If a modeller was to predict the company’s revenue, she should consider the impact of the Order-To-Cash PPIs and incorporate those into the simulation.

A valid question at this point is the expected number of KPIs and PPIs, which have to be considered to create such ontologies. Mostly, BPs are standardized, but are still sometimes customised to cater specific requirements of particular companies, for instance, introduction of additional activities in the standard Order-To-Cash BP. The fundamental PPIs, such as end-to-end execution time or instance occurrence, still remain valid. The same holds for universal KPIs (revenue, sales volume, cost), which are employed in all companies to check the health status of the business. On the other hand, there are also KPIs, which are unique for each different company, e.g., ”number of orders for product X” or ”current stock of product Y”. Because companies are selling so many different products and services, one can not simply map each single product into one universal ontology. In our framework, we have introduced the notion of KPI/PPI classes. These classes are used as templates in the ontology. A good example for such a class in our ontology is *NumberOfOrders*, which acts as parent for each single ”number of orders for specific product” KPI. With this available ontology, all that is left for the modeller to create CLDs, is: Classify the current KPI/PPI and query the relationships of its parent class to other KPIs/PPIs from the ontology.

The introduction of a link between the operational and strategical level within our framework offers one huge benefit: It enables the modeller to drill-down from a high-level strategic view to the low-level operational view. Since an enhanced CLD now contains both, the KPIs and PPIs, possible strategic KPI bottlenecks can be tracked all way down to the operational level, thus showing the root cause of deviations. This KPI-PPI connection is established with the help of ontologies within our framework. Additional benefits of having such an ontology is, that these are highly extensible and reusable for the targeted domains, for instance, any newly observed domain specific dependencies can be included into the ontology. On top of that, ontologies are well suited for automated processing and can easily be queried with SPARQL to retrieve the KPI/PPI relationships (The W3C SPARQL Working Group, 2013).

5 CONCLUSION AND FUTURE WORK

In this paper, we have proposed an advanced business simulation framework that addresses the missing connection between the strategic and operational level in businesses, thereby converging the two domains. Within our framework, the operational level PPIs are included as a part of BD simulation for KPI predictions and enabling operational level visibility. This means any KPI violation at the strategic level can be tracked down to the operational level to carry out corrective actions. Furthermore, in order to address the increased complexity resulting from the introduction of PPIs into BD simulations, we have designed and implemented additional SKMs. These SKMs (ontologies) provide knowledge about relations and dependencies of KPIs and PPIs, and can be used as *business dictionaries* to look up relationships of KPIs and PPIs either manually or automatically. An automated consumption of SKMs can further be used to automate the BD simulation process for continuous KPI predictions in real-time. So far this approach is unidirectional, because we have only incorporated PPIs as main influencing factor towards KPIs. For future work, it would be interesting to analyse the KPI influence towards PPIs as well. We believe, that the prediction of PPIs could also be beneficially impacted by incorporating the simulated KPIs. The next step would be, to evaluate the framework in the context of an industrial use-case, to demonstrate improved KPI predictions.

REFERENCES

Ann, R., Chase, G., Omar, R., Taylor, J., and von Rosing, M. (2011). *Applying Real-World BPM in an SAP Environment*. Galileo Press, Bonn.

Binder, T., Vox, A., Belyazid, S., Haraldsson, H. V., and Svensson, M. (2004). Developing System Dynamics models from Causal Loop Diagrams. Technical report, University of Luebeck, Germany; Lund University, Sweden.

Brockwell, P. J. and Davis, R. A. (2006). *Time Series: Theory and Methods*. Springer, second edition.

Burns, J. R. (1977). Converting signed digraphs to Forrester schematics and converting Forrester schematics to Differential equations. *IEEE Transactions on systems, man, and cybernetics*, 10:695–707.

Del-Rio-Ortega, A., Resinas, M., and Ruiz-Cortes, A. (2010). Defining Process Performance Indicators : An Ontological Approach. *On the Move to Meaningful Internet Systems: OTM 2010*, 6426:555–572.

Drobek, M., Gilani, W., and Soban, D. (2013). Parameter estimation and equation formulation in Business Dynamics. In *Third International Symposium on Business Modeling and Software Design*, Noordwijkerhout. ScitePress.

Ford, A. (1999). *Modeling the environment: An Introduction to System Dynamics Models of Environmental Systems*. Island Press, Washington, D.C.

Ford, D. N. and Sterman, J. D. (1998). Expert knowledge elicitation to improve formal and mental models. *System Dynamics Review*, 14(4):309–340.

Forrester, J. W. (1961). *Industrial Dynamics*. MIT Press; currently available from Pegasus Communications; Waltham, MA, Cambridge, MA.

Forrester, J. W. (1991). System Dynamics and the Lessons of 35 Years. pages 1–35.

Fritzsche, M., Picht, M., Gilani, W., Spence, I., Brown, J., and Kilpatrick, P. (2009). Extending BPM Environments of Your Choice with Performance Related Decision Support. In *Business Process Management*, pages 97–112. Springer.

Granger, C. W. J. (1969). Investigating Causal Relations by Econometric Models and Cross-spectral Methods. *Econometrica*, 37(3):424–438.

Hecking, M. and Schroder, C. (2013). *Current Implementation Level of Business Process Management in Corporate Practice: A Quantitative Analysis*. GRIN Verlag.

Heilig, B. and Möller, M. (2014). *Business Process Management mit SAP NetWeaver BPM*. Galileo Press Gmbh, 1st edition.

Howson, C. and Newbould, E. (2012). *SAP BusinessObjects BI 4.0 The Complete Reference 3/E*. McGraw-Hill Osborne, 3rd edition.

Intalio (2013). BPMS designer; http://www.intalio.com/products/bpms/overview/.

Ko, R. K. L., Lee, S. S. G., and Lee, E. W. (2009). Business process management (BPM) standards: a survey. *Business Process Management Journal*.

Porzucek, T., Kluth, S., Fritzsche, M., and Redlich, D. (2010). Combination of a Discrete Event Simulation and an Analytical Performance Analysis through Model-Transformations. In *IEEE ECBS*, pages 183–192.

Redlich, D. and Gilani, W. (2011). Event-Driven Process-Centric Performance Prediction via Simulation. In *BPM Workshops*.

Richmond, B. and isee systems (Firm) (2008). An Introduction to Systems Thinking: STELLA Software.

Robinson, S. (1964). *Simulation - The Practice of Model Development and Use*. John Wiley & Sons.

SoftwareAG (2014). Software AG: webMethods, last accessed: April 2014; http://www.softwareag.com/corporate/products/wm/bpm/overview/default.asp.

Steiner, T., Verborgh, R., Troncy, R., Gabarro, J., and Walle, R. V. D. (2012). Adding Realtime Coverage to the Google Knowledge Graph. In *Proceedings of the ISWC 2012*.

Sterman, J. D. (2000). *Business Dynamics: Systems thinking and modeling for a complex world*. McGraw-Hill, New York, NY.

The W3C SPARQL Working Group (2013). SPARQL Query Language for RDF; http://www.w3.org/TR/rdf-sparql-query/.

Van Der Aalst, W. (2011). *Process Mining - Discovery, Conformance and Enhancement of Business Processes*. Springer.

Woods, D. and Word, J. (2004). *SAP Netweaver for Dummies*. Wiley Hoboken.

Zhang, L. (2002). *Knowledge Graph Theory and Structural Parsing*. Ph.d. thesis, University of Twente.

Multi-Level Business Modeling and Simulation

Koen Casier, Marlies Van der Wee, Sofie Verbrugge
Internet Based Communication Networks and Services research group –IBCN, Ghent University, Belgium
Koen.casier@intec.ugent.be

Heritiana Ranaivoson, Tanguy Coenen, Camille Reynders
SMIT research group, Vrije Universiteit Brussel, Belgium

Keywords: Cost Benefit Analysis, Computer Networks, Computational Modeling, Graphical Models, Cost Benefit Modeling, Business Modeling.

Abstract: The rapid succession of technological advances leads to important convergences of applications, devices and networks. More and more firms, previously locked in a niche, are exposed to a more global market and interactions with other firms. Pushing a new offer on the market requires a thorough understanding of this altered market. In essence, pushing a new offer requires basic business modeling and simulation. Often, this is performed by making a “back of the envelope” calculation. This calculation quickly grows out of proportions if the novel business proposition requires interactions with many other parties. In this paper, we present a scalable multi-level business modeling and quantification approach. It combines the intuitive structure and interactive discussions of a multi-user business modeling tool, while directly linking to a lower level for more technical modeling and simulation of costs and revenues. By combining these two levels of refinement, the business aspects are clearly separated from the calculation techniques, increasing ease and speed of modeling at the business side. Delegating the cost calculations to the more technical models allows for a truthful and reliable mimicking of the actual structure and costs. To achieve this, several detailed cost modeling languages are presented and linked to the higher level business modeling. Finally, this multi-level business modeling and simulation approach is applied to the case of an open access fibre to the home network deployment. The results clearly show the power of using such a multi-level business modeling and simulation approach.

1 INTRODUCTION

Increasing competitive pressure makes business model innovation an important issue for most companies. Especially collaborative business models, which require a strategic fit between various stakeholders involved, require an intensive interaction and consensus building related to assumptions, architectures and outcomes. Like many other creative processes (architecture, software design, new product development, etc.), business model innovation can also be supported by tools (Coenen et al, 2010). However, while a few basic frameworks exist, tool development for collaborative business model innovation is still in its infancy.

Many business modelling approaches start from a conceptual visualization of the context of one offering on the market, typically of one firm (Osterwalder et al, 2005) and (Al-Debei et al, 2010). The extended notion of a multi-firm interaction with several offerings and objectives exists and is typically visualized by means of a value network (Pijpers 2011). Still, most approaches are aimed at visualizing the interactions, often around one central firm and looking at one final offering on the market. A more truthful representation of reality in a business model should be seen as a network or graph of actors, the activities they perform, and all kinds of interactions between these actors. In order to be useful in advanced analysis, such business model should be set up according to a standardized ontology for which we used the SIMBU method (Coenen et al, 2009) as a starting point. The simple but expressive ontology proposed there, allows for the creation of complex business models and permits the support of consecutive simulations, to give users a better basis for decision making.

Although a business model consists of smaller elements as roles, actors, activities, value streams, etc., this is much too high level to estimate costs, revenues, losses or profits. Traditional cost benefit analysis, on the other hand, takes a closer look at the economics of a new investment, starting from dedicated calculations (Analysys Mason, 2008) in isolated cases and working with generic and reusable cost modeling languages and calculations (Van der Wee et al, 2012). Making dedicated calculations requires every researcher to redo the modeling if they have no access to the original model. Additionally, it does not allow extending and linking to other models. Building complex models will benefit from reusability, verification, exchange between researchers and business experts and linking to other models. Additionally using domain specific languages as opposed to grand tools with many parameters, will increase the transparency and ease of reading and understanding. Combining both approaches by working with standardized, reusable, domain specific languages will increase the strength of the business models.

On the one hand business experts (e.g. CEOs, entrepreneurs) talk about the roles and interactions of the different actors in a business model when they want to introduce a novel product or service on the market. On the other hand, a cost-benefit analysis is typically built for an isolated business case (one actor only), using dedicated, purpose-built models. If both approaches can be captured with the right level of detail and domain specific intuitive models and linked to each other, this will lead to additional information on the full business model, as well as on the isolated business cases. The combination of approaches will allow business experts to work on a higher level and design the business model as links between more detailed cost-benefit models, e.g. cloud infrastructure, network installation, etc. These models are then delegated to technical experts and more detailed modeling languages. A repository of models and fragments at both levels will increase the applicability of the approach and the speed of prototyping business models. This paper presents the combined approach, which is under active research and development and is called hereafter the BEMES (Business Modeling and Simulation) approach.

In this paper, the BEMES approach is applied to a prototype business model for a fiber to the home (FTTH) case, where one physical infrastructure FTTH provider is installing a new FTTH network and opening up this network in a non-discriminatory way to all available network and service providers. The multi-level business modeling approach allows visualizing the main business interactions rapidly, and learns about the profitability of all actors at the same time. It also shows the ways one firm's failing business case can be made viable within the group of actors in the full business model.

In section II, the BEMES business modeling tool is rapidly introduced and then compared to some of the main existing business modeling approaches.

As mentioned above, the business models need to be complemented with a cost-benefit analysis in order to get correct and useful advice and information from the business model. Building a reliable cost-benefit analysis also benefits from using problem specific modeling languages. In Section III, an overview of existing and novel cost modeling languages (technical expert tools) is presented.

In Section IV, both levels of modeling are linked to each other. As a proof of concept, a multi-level model for an open access FTTH network is built and the results of this model are inspected.

Finally, Section V concludes by summing up the main findings of this work and by presenting future steps in the development and extension of this multi-level business modeling and simulation approach.

2 BUSINESS MODELING

People use business modeling with the aim to analyze the current functioning of a firm or an industry, identify challenges, and possibly propose better business configurations. When building the business model, users need a highly interactive tool for drawing and discussing on their view of the industry actors and their interactions. It should be sufficiently high level, and no detailed cost and revenue discussions or simulations should be necessary at this level. The BEMES Business Modeling, proposed in this paper, is based on the SIMBU method (Coenen et al, 2009). It features a value-flow based approach, and uses a simple and intuitive ontology specifically designed to allow for collaborative business modelling.

With BEMES, building a business model consists in identifying every actor, their activities and the interactions between their activities. A business model configuration corresponds to a given business model with specific values (e.g. cost amounts, revenue percentages, etc.). It is easy to compare different configurations or scenarios by playing with these values within the given business model. It is also possible to compare different approaches in setting up a working business model by playing with the definition of the actors, the definition and repartition of the activities, and the definition of the relationships between the activities.

The following elements are required to get a both intuitive limited modeling set and a high expressivity of the model:

- The Actor represents a business model stakeholder.
- An Activity is undertaken by an actor.
- A Flow represents a relationship between Activities.
- A flow can either be a monetary flow or a flow of goods and/or services.
- Value Sharing represents the division of one monetary flow into multiple monetary flows.
- A Swim lane is horizontally oriented and groups all the activities of an actor.
- A sub model allows the business modeling to be itself hierarchically structured and is typically used to increase readability of the model.

In Figure 1, a FTTH network is modeled, using BEMES. There are three Actors in this Business Model: (1) the Customer who buys network connectivity at (2) the Network Provider. This Network Provider, in order to be able to provide network connectivity needs (3) the Physical Infrastructure Provider, who provides physical network and connects customers. Arrows between the Activities performed by each Actor show the flows of money or services between the Actors' Activities.

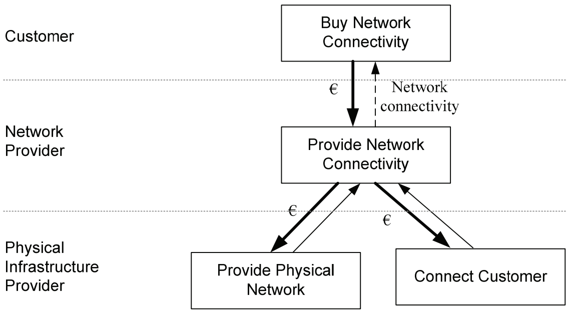


Figure 1: business model for the open access FTTH deployment.

2.1 Comparison of BEMES to other Business Modeling

Various other business modeling approaches and languages exist and Table 1 provides a comparison of BEMES to the mainly used other business modeling approaches. Every approach's main advantages and disadvantages are briefly discussed after the table.

One of the most salient business modelling approaches is the Business Model Canvas (BMC), based on Osterwalder's work (Osterwalder and Pigneur, 2010). The Business Model Canvas is an ontological construct composed of 9 different categories participants (key partners, key activities, value proposition, customer relationships, customer segments, key resources, channels, cost structure and revenue streams) that need to be reflected on by a group of stakeholders. In the Business Model Canvas philosophy, a brainstorming session is done as a workshop, where all participants are asked to place Post-it notes on the canvas and discuss the implications of their actions.

Table 1: Comparison of business modeling approaches.

	BEMES	e^3 value	Moby	BMW	BMC	CBM
Value Proposition	+/-	- -	+ +	-	+ +	+ +
Multiple Actors	+ +	+ +	+ / -	- -	- -	- -
Flexible Relations	+ +	+ +	+ +	- -	- -	- -
Value Net Completeness	+	+ +	+ +	- -	- -	- -
Value Net Simulation	+ +	- -	+ / -	+ / -	- -	- -
Ease of Use	+	- -	- -	+ +	+ +	+
Intuitivism	+	-	-	- -	+ +	+ +

Konnertz (Konnertz et al, 2011) has proposed the collaborative business modelling (CBM) approach, which uses the Business Model Canvas by placing post-it notes on the canvas, generating a number of business models. After this is done, the business models are prioritised on the dimensions of attractiveness and effort. The models that are most attractive and take least effort are the ones that get most attention in the validation phase.

The Business Model Canvas method can be a powerful eye-opener and a good brainstorming framework, but it has some severe limitations if one wants to use it in an Open Business Model Innovation process. Firstly, the output of the Business Model Canvas method is a list of elements that can be bundled in scenarios. There is little support for making plain the relationships and interactions between the different elements. These relations and interactions make the difference between a business model as a static list of its constituents and a business model that is dynamic, as is the environment in which it will operate.

In terms of the open innovation perspective (Chesbrough, 2005), some categories exist in the Business Model Canvas approach that can be linked to more open, multi-actor, value-networks, like key partners and channels. Still, the reflection engendered by the Business Model Canvas is mainly focused on one organization. In Open Business Model Innovation, several business actors collaborate to realize a value proposition in a relationship of mutual benefit. Finding a sustainable business model requires the perspective of the different actors to be made explicit and combined in a consensus business model.

Second, the Business Model Designer, described in (Weiner and Weisbecker, 2011), is built on a very broad ontology and allows the creation of very detailed representation of the components that are related to a value proposition, both within an organisation and outside of it. In particular, it allows the mapping of resources that an organisation should use in order to realise the value proposition, as well as the competitors that the organisation will have to deal with.

Third, the Business model wizard (BMW) allows for the creating of a business model by configuring 25 elements using an online form. The result is a business model that can be analysed and compared to the business models of existing organisations. While this is an easy to use approach, it focuses on one organisation and is constrained by the 25 elements that are part of the model.

Fourth, the e3-value modelling approach proposed by (Gordijn et al, 2011) has tool support in the form of the e3 editor. This approach focusses more on the dynamics of the business model than on its constituents. The e3-value approach allows for the creation of highly formalized business models. However it requires a substantial amount of time in order to learn the interface and the modelling language.

In conclusion, BEMES is simple yet powerful. It allows for the expression of a complex business model, while being easy to learn by the modelers. The emphasis is more on the business knowledge of the modeler than on the business modeling skills of the modeler. Furthermore, the business modeling ontology allows for easy understanding of models created by others, which supports collaborative business modeling. Finally, models built using this ontology can be used to do high-level or detailed quantitative cost and revenue simulations.

3 COST MODELING

When making a business plan for the deployment of a novel open access FTTH network, close interaction between the physical infrastructure provider, network provider and any other involved parties will be necessary. The different parties will especially be interested to learn more on the costs they incur and have to pay to the other parties and to what final customer price this will lead. The FTTH network consists of the outside plant as well as the in-house installation and the installation of all central office equipment. It also requires operational processes in order to keep the network up and running and to sell services on top of this network. Much literature exists on how to build a business case for an FTTH network (e.g. Analysys Mason, 2008, Van der Wee et al, 2012, Banerjee and Sirbu, 2003 or Medcalf and Mitchell, 2008). However it is hard for a researcher to follow the model in all these papers, as the models are typically not expressed in a format easy to read, easy to duplicate and use in other modeling steps. Additionally, each paper has a separate focus and only includes certain parts of total costs, making comparison very difficult.

Visual modeling languages help in making the business modelers and technical experts quickly aware of what it takes into account and how it will be calculated. A uniform and consistent translation from model to costs furthermore assures a correct calculation of the costs and deduces the right economic indicators from it. In what follows we link existing modeling approaches to each other and apply them in making a solid cost model for open access network architectures. Where necessary, we introduce and developed a novel modeling approach and link to more rigid specifications of the new language. Three cost modeling approaches are presented – infrastructure or network modeling, equipment coupling modeling and operational modeling – and one novel language for modeling the way revenues are estimated. The former three will be directly mapped to an activity on the business model, where the latter is linked to the monetary arrows linking these activities. All four combined allow rapid and reliably estimating the investment costs and revenues of the business case at hand.

3.1 Infrastructure Modeling

The largest cost of the network will come from installing the outside plant, i.e. deploying the fiber into trenches to connect all customers to the central office. Several models already exist for making an analytical estimation of this cost (Mitcsenkov et al, 2013). Considering the size of this cost, a more detailed calculation can be made using an ILP formulation. In (Mitcsenkov et al, 2013) a comparison is made between two analytical models and a full optimal installation calculation tool, and the street based estimation model will be used in the following example calculations. As an example area we use the city of Ghent, the third largest city of Belgium counting almost 235,000 inhabitants on an area of 156 km². The FTTH rollout is limited to the city center, with ca. 90,000 inhabitants or 42000 families on 20 km². (Gent, 2013).

3.2 Equipment Modeling

The second important cost is linked to the installation of the equipment in the central office. In order to calculate the costs of the installation of this equipment and taking into account all possible failures of this equipment in time and their replacements, we developed a novel modeling format. This modeling format is based upon previous work (Van der Wee et al, 2008), and extended with indications of replacement period of the equipment (in accordance to either proactive maintenance or of failure rates), power consumption, floor space consumption, etc.

The model is based on (1) main drivers for equipment installation which are represented by arrows and will be used in the calculation of the required amount of equipment linked to these drivers. Every block from thereon will become a driver for next blocks once calculated. (2) Equipment blocks that hold all information on the cost, replacement time, etc. and (3) aggregators which will aggregate the incoming demands from different drivers and sub-equipment in a specific manner (sum, max, etc.). Finally all blocks are linked to each other by means of lines with an aggregation factor. More information on this novel format can be found at (Casier, 2013). The equipment model used in this simplified business model is shown in Figure 2.

3.3 Operational Modeling

Operational modeling is based on the standardized Business Process Modeling Notation (BPMN) (OMG, 2013) restricted to a smaller subset only containing the main flowchart structures required for cost calculation. Using an approach based on activity based costing (Kaplan and Anderson, 2004) and described in (Casier, 2009), the costs can be linked to the execution of the process for a given planning horizon. The model for customer connection used in the case of an open access FTTH network is shown in Figure 3.

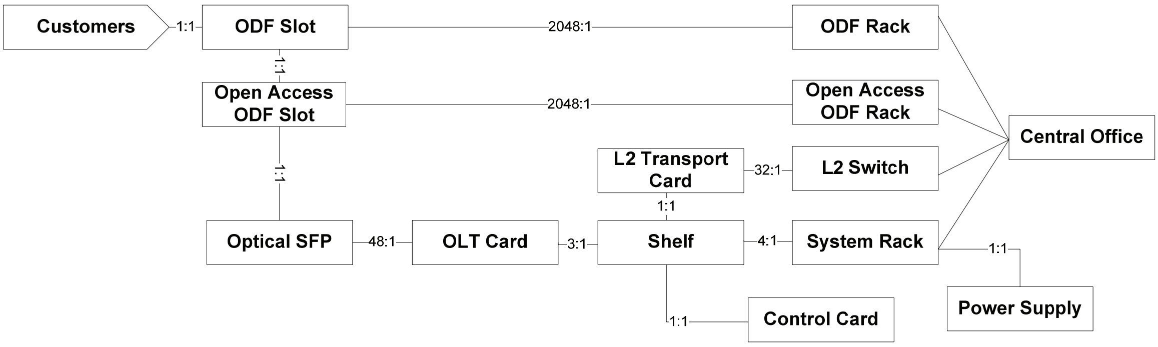


Figure 2: Equipment Coupling Modeling Notation for an open access central office infrastructure and network installation.

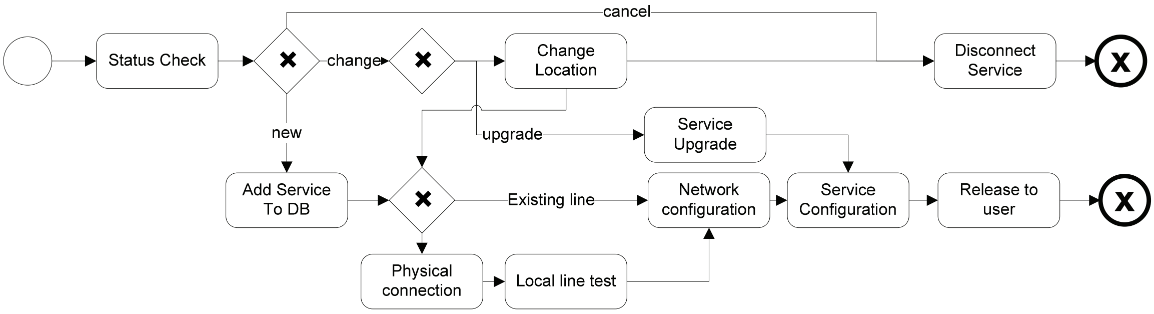


Figure 3: Business Process Modeling Notation for the operational process of customer provisioning.

3.4 Revenue Modeling

Once the full network, the equipment and the operational expenditures are modeled, all costs of the business model can be calculated. Still, this is only part of the analysis and should be complemented with a modeling of the revenues for each activity in the business model. The revenue modeling is aiming to calculate the revenues based on the costs and the number of paying customers. As such it has a notion of a fixed revenue scheme but also of an adaptive scheme aiming at break-even or a profit over break-even with an adjustable timing on when to get up to this point. Additionally the revenue model allows switching between revenue schemes at a given time or condition (e.g. critical customer mass reached). This revenue modeling allows answering questions on the main economic indicators such as profitability, minimal and advised pricing or payback period. A more formal specification of the full model is given in (Casier, 2013). The models used in pricing the open access and the final connection price are kept deliberately very simple, where we assume each role to aim for 20% profit over a planning horizon of 10 years. Deploying the physical infrastructure of the network will make an exception to this and aim for 10% profit over a planning horizon of 20 years.

4 MULTI-LEVEL MODELING

The multi-level modeling links the cost and revenue models to the respective higher level elements, activities and monetary flows in the business model. A cost estimation model is attached to each activity in the business model actually leading to costs in its execution. A revenue model is attached to each monetary flow between two activities. Finally, additional input (e.g. amount of customers or price of equipment) can be defined as time-dependent values and linked to the models of the activities. Once all inputs are defined, all activities causing costs and all monetary flows are linked to a cost, respectively a revenue model; the business model can be fully simulated.

The calculation starts from the activities of the graph which have no outgoing monetary value exchanges, or in other words, which use no service from a lower level activity for which they are charged. The costs in these blocks can be fully calculated using their internally attached cost model. In the case of Figure 1 the cost for the physical infrastructure can readily be calculated. When this cost is known together with the expected amount of customers, executions, etc., this should be charged to the revenue model and linked to the monetary incoming arrow(s) to be able to calculate pricing and total revenues. Again in Figure 1 the amount the PIP will charge to the NP for the use of its infrastructure can now be calculated. The same calculation steps can be taken for connecting the customers and the price charged for this role to the NP. At this point the network provisioning is becoming the next point in the calculation, as all outgoing value exchanges linked to this activity are fully quantified. And finally this allows calculating the price to charge to the end customer. This recursive scheme allows all activities and monetary flows in any business model to be fully quantified.

We translated Figure 1 into a business model configuration that can be simulated by attaching the infrastructure cost model to the physical infrastructure deployment role, linking the equipment model for the active equipment to the network deployment role and finally operational model to the network provisioning role. We assume all monetary flows to aim for 20% profit on the costs of the role (and underlying roles). As mentioned the infrastructure considers only 10% profit. Additional information can be exchanged between models, as for instance the amount of installed equipment will be the driving value for operational maintenance.

When calculating the business model for the given scenario, the different cost components are calculated in terms of the amount of customers in the area (physical infrastructure) and the amount of customers to connect to the network. The first is equal to the amount of inhabitants in Ghent and for the second, we consider a bass adoption curve with as market potential 95%, with innovation (p) 0.03 and imitation (q) 0.38. A demand aggregation of 30% is expected as a boundary condition for the FTTH network deployment. All costs of the physical infrastructure and network provider are discounted with a discount factor of 5% respectively 10%.

Figure 4 gives the results for the cumulative discounted costs, revenues and outcome for the 10 years for (top) the PIP infrastructure, (middle) the PIP operational expenditures and (bottom) the overall NP outcome.

In this business case the PIP will have to charge a price of €235 per customer per year to the NP for the use of its infrastructure and a price of €42 for

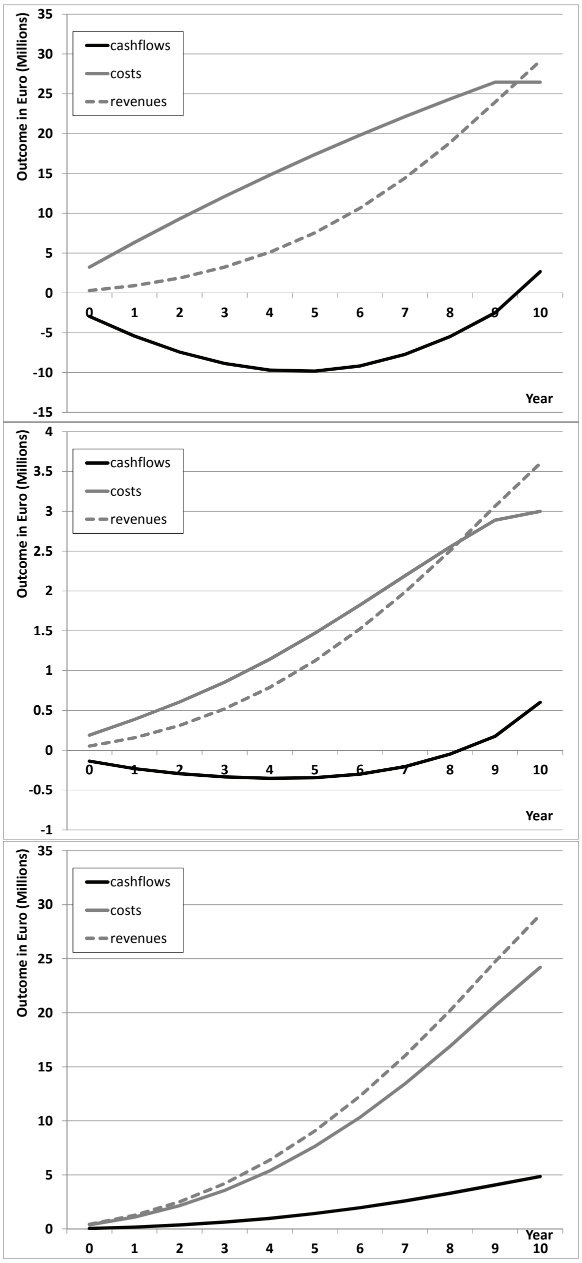


Figure 4: Overview of the costs, revenues and profit for the different roles in the open access business case.

making a connection to a customer. This already incorporates the fact that customers will only need a one-time physical connection and changing providers afterwards does not require dispatching an installation team. This leads to an overall cost of €277 per customer per year charged from the PIP to the NP, which is equal to a monthly price of €23 (€19.5 for the infrastructure).

The NP will additionally provide the necessary equipment and make a contract with the PIP. In order to accomplish this, the NP needs to charge the customer a yearly fee of €339 or a monthly fee of approximately €28.

5 CONCLUSIONS & FUTURE WORK

Building a viable business case for a commercial offering based on novel technology on the market is not straightforward; especially in case different actors have to cooperate. Estimating the viability of such business cases requires input and knowledge from two research fields – (1) techno-economic research in which cost simulation models are built and (2) business modeling in which graphical models are focusing on the roles, actors and their interactions. A combination of both requires a multi-level business modeling approach in which a graphical business model is linked to separate techno-economic simulations. Clearly this will require an intuitive and complete business modeling ontology in combination with domain specific techno-economic cost as well as revenue simulation languages.

In this paper we have presented a multi-level business modeling approach – called BEMES – with a very intuitive yet complete business modeling ontology and linked (in an extensible manner) to network infrastructure, business process and a novel equipment modeling as well as to a (also novel) revenue modeling. We have used this approach to build a business model for an open access FTTH network deployment in which a physical infrastructure provider is leasing the fibers to a network provider together with the operations for connecting customers. Both actors will aim for a profit of 10% (infrastructure) respectively 20% (network). This business model configuration clearly shows the value of BEMES as the viability of the overall business case can be quickly checked against the final subscription price that needs to be charged to the customers. In this way we learn that an open access deployment in the city center of Ghent should demand at least €28 for providing FTTH connectivity to the end customer.

ACKNOWLEDGEMENTS

The authors have received funding from the Agency for Innovation by Science and Technology in Flanders, Belgium, and the BEMES project is part of the FI-WARE: Future Internet Core Platform European project (grant-nr. 285248).

REFERENCES

Al-Debei, Mutaz M., Avison D.. "Developing a unified framework of the business model concept." European Journal of Information Systems 19.3 (2010): 359-376

Analysys Mason, "Final report for the Broadband Stakeholder Group: The costs of deploying fibre-based next-generation broadband infrastructure", Ref: 12726-371 (8 September 2008).

Banerjee A. and Sirbu M., "Towards Technologically and Competitively Neutral Fiber to the Home (FttH) Infrastructure," in Proceedings of The 31st Research Conference on Communication, Information and Internet Policy, Washington DC, USA (19-21 September 2003).

Casier K, Techno-Economic Evaluation of a Next Generation Access Network Deployment in a Competitive Setting, PhD degree at the Faculty of Engineering of the Ghent University obtained, October 8th 2009

Casier K, Revenue Modeling Wizard Specification, 2013, http://forge.fi-ware.eu/plugins/mediawiki/wiki /fiware/index.php/FIWARE.OpenSpecification.Apps. BusinessCalculator.Revenue

Casier K., The Equipment Coupling Modeling Notation (ECMN) Specification, 2013, http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/ FIWARE.OpenSpecification.Apps.BusinessCalculator .ECMN

Chesbrough, H. W. "Open innovation : the new imperative for creating and profiting from technology" Boston, Mass.: Harvard Business School Press., 2005

Coenen, T., De Waele L., Ballon P.. "The Simbu method-Supporting creativity in business modelling for open innovation" 2009

Coenen, T., Ranaivoson, H., Ballon, P. "Tool support for the Open Business Model Innovation funnel". Proceedings of ISPIM Innovation Symposium 2010, Quebec, Canada.

Gent, Gent in Cijfers, Online information on the city of Ghent (Dutch), http://gent.buurtmonitor.be/

Gordijn, J., De Leenheer P., and Razo-Zapata I. "Generating Service Value Webs by Hierarchical Configuration: A Case in Intellectual Property Rights Clearing." System Sciences (HICSS), 2011 44th Hawaii International Conference on. IEEE, 2011

Kaplan R.S., Anderson S.: "Time–Driven Activity-Based Costing," Harvard Business Review, 2004, vol. 82, no. 11, pp. 131-138.

Konnertz, L, Rohrbeck R., and Knab S. "How collaborative business modeling can be used to jointly explore sustainability innovations."ISPIM Annual Conference, Hamburg, Germany. 2011.

Medcalf R., Mitchell S., "Fiber to the Home: Technology Wars", http://www.cisco.com/web/about/ac79/docs/wp/Tech_ Wars_Part__1_Ext_EI_0128.pdf (January 2008).

Mitcsenkov A., et al., Geometric versus Geographic Models for the Estimation of an FTTH Deployment, Telecom Systems Journal special issue, May 2016 (accepted)

Object Management Group/Business Process Management Initiative, http://www.bpmn.org/

Ostenwalder A., Pigneur Y., Tucci C.L., Clarifying Business Models: Origins, Present, and Future of the Concept, Communications of AIS, Volume 15, Article 1, 2005

Osterwalder, A. & Pigneur, Y., 2010. Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers, Hoboken, New Jersey: John Wiley & Sons

Pijpers V. • De Leenheer P., • Gordijn J.,• Akkermans H., Using conceptual models to explore business-ICT alignment in networked value constellations, Requirements Engineering (13 October 2011), pp. 1-24

Van der Wee M., et al., A modular and hierarchically structured techno-economic model for FTTH deployments. Comparison of technology and equipment placement as function of population density and number of flexibility points, ONDM 2012

Weiner N, Weisbecker A. "A Business Model Framework for the Design and Evaluation of Business Models in the Internet of Services." SRII Global Conference (SRII), 2011 Annual. IEEE, 2011

A Component Abstraction for Localized, Composable, Machine Manipulable Enterprise Specification

Vinay Kulkarni[1], Tony Clark[2] and Balbir Barn[2]
[1]Tata Research Development and Design Centre, Tata Consultancy Services, 54B Hadapsar Industrial Estate, Pune, India
[2]Middlesex University, London, U.K.
vinay.vkulkarni@tcs.com, {t.n.clark, b.barn}@mdx.ac.uk

Keywords: Enterprise Modelling, Component.

Abstract: Enterprise modelling aims to specify an enterprise in terms of high-level models that address key problems such as business-IT alignment, enterprise transformation and optimal operation. No two situations in real world enterprises are exactly alike but there may be significant overlap. Relative ignorance of such overlaps forces essentially the same problem, albeit in a different context, to be repeatedly solved from scratch. This is a time-, effort- and cost-intensive endeavour. To overcome this problem and facilitate reuse, we propose a model-centric component abstraction that enables specification of the *what*, the *how* and the *why* concerns of enterprise in a localized, composable and machine manipulable manner. We present a meta-model, describe concrete syntax for its textual representation, and discuss the required model processing machinery.

1 INTRODUCTION

Modern enterprises operate in a highly dynamic environment wherein changes due to a variety of external change drivers require rapid responses within a highly constrained setting. This calls for precise understanding of: *what* is the enterprise, *how* it operates and *why* it so operates, the set of change drivers, a set of possible to-be states, and a quantitative and qualitative to-be state evaluation criteria. Understanding is typically required at several levels of granularity: a department, a business unit, the entire enterprise *etc*. The scale of a modern enterprise means this understanding exists only for highly localized parts and typically in the form of documents or descriptive models (Zachmann 1999, TOGAF). Popular enterprise modelling tools, *e.g*., ArchiMate (http://www.visual-paradigm.com) offer little support for the quantitative or qualitative analysis of enterprise models. As a result, experts are forced to rely solely on their experience when faced with a specific problem. Thus, fractured incomplete knowledge and sole reliance on human expertise emerge as the principal contributing factors leading to change responses that are inaccurate, inefficient and ineffective.

Key decision-makers in enterprises face generic problems: business-IT alignment, optimizing cost of IT to business, transformation with certainty *etc*. These problems manifest differently in different contexts and yet share significant commonality. For instance, the details of wealth management bank merger differs on a case by case basis; however cases have much in common both in terms of problem formulation as well as solution. Current Enterprise Modelling state-of-the-art as well as of-practice completely ignores this commonality. As a result, each problem instance needs to be solved afresh. This is a highly cost-, time-, and effort-intensive endeavour.

The problems we seek to solve are: (i) to reduce excessive dependence on human experts for decision making, (ii) to address the commonality across different instances of a generic problem and, (iii) to address scale and complexity.

Our proposition addresses the problems as follows: (i) the *what*, *why* and *how* perspectives of an enterprise are captured in terms of a core conceptual meta-model, (ii) DSLs and patterns are used to capture reusable knowledge and translated in terms of the core concepts into a kernel language, (iii) the kernel language supports features to address scale and complexity including composition, encapsulation, and higher-order features.

Our approach is extensible through the use of a plug-in architecture supporting DSL-based models translated to an executable kernel language defined in terms of a core collection of concepts. Section 2 discusses related work. Section 3 provides an overview of the different aspects of the approach and section 4 concludes by describing our progress to date and our future directions.

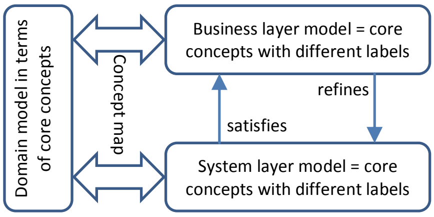

Domain model in terms of core concepts
Concept map
Business layer model = core concepts with different labels
refines
satisfies
System layer model = core concepts with different labels

Figure 1: Purposive meta-models.

2 RELATED WORK

The current state-of-the-art of enterprise modelling (or specification) can be broadly classified into two: those that focus on *what* and *how* aspects (Clark et al., 2013, Zachmann 1999, TOGAF, Wisnosky and Vogel, 2002) and those that focus on *why* (Yu et al., 2006, Dardenne et al., 1993, OMG BMM 2010). The supporting infrastructure for the former, with the exception of the ArchiMate tool (bit.ly/1s1WyTv) is best seen as a means to create high level descriptions for human experts to interpret in the light of synthesis of their past experience. The Stock-n-Flow model (Meadows 2008) provides a different paradigm for modelling *what* and *how* aspects and comes with simulation machinery for quantitative analysis (http://bit.ly/1hebMvC). Several BPMN tools providing simulation capability exist but are limited to the *how* aspect (http://bit.ly/PdkqVg). Supporting infrastructure for *why* (http://bit.ly/Oav0Lm) is comparatively more advanced in terms of automated support for analysis. Informed decision making demands taking into account all the three aspects in an integrated manner, however, correlating *what* and *how* with *why* remains a challenge. Given the wide variance in paradigms as well the supporting infrastructure, the only recourse available is the use of a method to string together the relevant set of tools with the objective of answering the questions listed earlier. The non-interoperable nature of these tools further exacerbates automated realization of the method in practice. As a result, enterprises continue to struggle in satisfactorily dealing with critical concerns such as business-IT alignment, IT systems rationalization, and enterprise transformation.

3 PROPOSED SOLUTION

We propose a modelling language engineering solution based on the principles of separation of concerns (Tar et al., 1999) and purposive meta-modelling. We posit a core language defined in terms of generic concepts such as *event*, *property*, *interface*, *component*, *composition,* and *goal*. They constitute a minimal set of concepts necessary and sufficient for enterprise specification. The core language can be seen as a meta-model template where the generic concepts are placeholders. In the proposed approach a template emits the desired purposive meta-model through a process of instantiation wherein the placeholder generic concepts are replaced by purpose-specific concepts. This makes it possible to establish relationships across multiple purposive meta-models as shown in Fig. 1 and also impart consistent semantics.

The meta-modelling approach is suited to the open-ended problem space of enterprise modelling: any number of meta-models can be defined, relationships spanning across the various meta-models specified and the desired semantic meaning imparted *etc*.

3.1 Component Abstraction

A component is a self-contained functional unit with high coherence and low external coupling. A component exposes an interface stating the externally observable goals, expectations from the environment, mechanisms to interact with the environment, and encapsulates an implementation that describes how the exposed goals are met. A component can make use of several contained components in order to meet the promised goals. A component participates in hierarchical composition structure to accomplish wider goals of the enterprise, *e.g.,* larger unit or an enterprise. The expectations of a component from its environment are accompanied by a quality of service guarantee and together both constitute a negotiating lever. Thus, a component is in fact a family of (member) components where all family members have the same goal and interaction specifications but differ only in terms of the quality of service delivered and the expectations from the environment for delivering the promised quality. The core concepts of a component abstraction are depicted in Fig. 2.

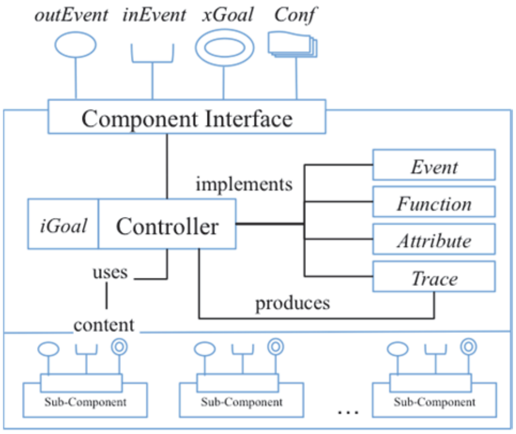


Figure 2: Component abstraction.

We draw from a set of existing concepts to derive the component abstraction. *Modularization*, *reflective component hierarchies* and *interface-implementation separation* are taken from Fractal Component Models (Barros et al., 2009). *Goal-directed active behaviour traces* are taken from Agent Behaviour (Bonabeau, 2002). Defining *component state* in terms of *attributes* and *traces* is borrowed respectively from object oriented design patterns (Gamma et al., 1995) and *event driven* enterprise architecture modeling (Clark and Barn, 2011). The event driven architecture (Michelson 2006) is a means to support flexible interactions protocol between components. Finally the concept of *intentional modelling* (Yu et al., 2006) is adopted to enable specification of component goals.

3.2 Component Meta-model

The proposed component meta-model is depicted in Fig 3. A component has two parts – an interface and implementation: the interface addresses the *what* and *why* aspects, the implementation addresses the *how* aspect. Thus a component (C) is a tuple <*CI*, *Impl*>. A Component Interface(*CI*) is a tuple <*inEvent*, *outEvent*, *xGoal*, *Conf*> where *inEvent* is a set of events of interest to the component, *outEvent* is a set of events generated by the component, *xGoal* is the external observable goal of the component and *Conf* is a set of configuration variants that conform to the *InEvent*, *OutEvent* and *xGoal*. *Conf* is a tuple <*Expect*,*QoS*> where *Expect* is the set of expectations from the environment expressed as name-value pairs, and *QoS* is the set of *QoS* properties to be guaranteed by the component provided the expectations are met. Implementation (*Impl*) is a tuple <*iGoal*,*P*,*F*,*T*,*Content*,*iEvent*> where: *iGoal* is the internal goal of the component, *P* is a set of properties or attributes, *F* is set of functions each encoding a computation, *T* is a trace of past consumed and produced events, *Content* is a set of its sub-components, and *iEvent* is set of internal events used to interact with sub-components.

Component implementation is necessary and sufficient to cater to all its variants as specified in *Conf*. The control unit *CU* represents implementation of a component. It responds to *inEvent* set of events, raises *outEvent* set of events, and orchestrates the *Content* sub-components so as to accomplish the stated *iGoal*. It records the events of interest and changes to component properties (*P*). The control unit captures the behaviour of component *i.e.*, a set of handlers for all *inEvent*.

Event (*E*) is defined as tuple <*Name*,*EP*, *preCond*,*postCond*> where *Name* is the identifying label, *EP* is the set of properties or attributes of the event, *preCond* is the condition that must be fulfilled to recognize the event, and *postCond* is the condition that must hold true after completion of the event. The *preCond* and *postCond* are expressions over events and event properties.

Goal is a tuple <*Name*, *GExpr*> where *Name* is the identifying label and *GExpr* is either property expression (*PExpr*) or event expression (*EExpr*) or goal composition expression (*GCExpr*). Property expression is value expression over properties (*P* and *EP*), event expression is an LTL formula over events, and goal composition expression uses a set of composition operators over goal expressions. The goal composition expression enables specification of limited uncertainty and non-determinism in the goal.

3.3 Language Features

The approach outlined above relies on being able to represent and process an organisation that is expressed in terms of a component-based abstraction. We envisage a product-line approach (Reinhartz-Berger, 2013) whereby a suite of tools based on this abstraction is used to facilitate a collection of different organisation analysis and simulation activities. Each activity will constitute a *domain*, *e.g.*, cost analysis, resource analysis, mergers and acquisition, regulatory compliance. In principle, each new domain will require a new domain specific language to represent the concepts. How should such a proliferation of domains be accommodated by a single component abstraction?
Our proposal is to construct an extensible kernel language that is used as the target of translations from a range of domain specific languages (DSLs).

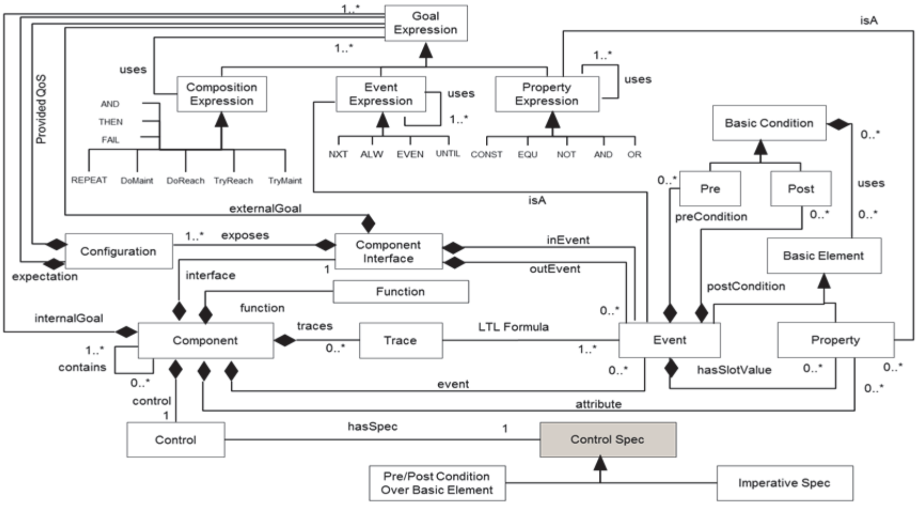


Figure 3: Component meta-model.

Each DSL supports an organization analysis and simulation use-case. We then aim to construct a virtual machine for the kernel language so that it is executable. Model execution supports organisation simulation and some analysis use-cases. Links to external packages such as model-checkers will complete the analysis use-cases.

The use of a single kernel language provides a focus of development effort and can help minimise the problem of point-to-point integration of analysis methods. Our proposal is that the concepts defined by the model in Fig. 3 are a suitable basis for most types of analysis and simulation use-case and therefore the kernel language will be defined in terms of these concepts.

Given its ability to accommodate multiple simulation and analysis use-cases, we envisage the language being the basis of a suite of organisational modelling, simulation and analysis tools, presented in the form of a single integrated extensible meta-tool. Since organisational information is likely to be very large (at least many tens of thousands of model elements) it is important the tool is efficient, scalable, supports distributed development and is flexible in terms of its architecture. To this end we aim that the language should be compiled to a machine language running on a dedicated VM, the language integrates with standard repository technology, and can run equally well on single machines, networked machines and via the cloud.

Organisations consist of many autonomous components that are organized into dynamically changing hierarchical groups, operate concurrently, and manage goals that affect their behaviour. We aim for the kernel language to reflect these features by having an operational semantics based on the Actor Model of Computation (AMC) (Hewitt, 2010) and its relation to organisations, or *iOrgs* (Hewitt, 2009). Actors have an address and manage an internal state that is private and cannot be shared with other actors in the system. Execution proceeds by sending asynchronous messages from a source actor to the address of a target actor. Synchronous messages can be achieved by sending an actor in an asynchronous message to which the result should be sent. Each message is handled in a separate execution thread associated with the target of the message and the message itself (collectively referred to as a *task*). During task-execution an actor may choose to change its state and behaviour (*becoming* a new actor) that is immediately available to process the next message sent to the target address.

Our claim is that the AMC provides a suitable basis for execution and analysis of the concepts defined in Fig. 3 and can be used to represent the features of a component. The rest of this section lists the key features that must be supported by the kernel actor-based language:

[**adaptability**] Organisational components may change dynamically during a simulation. Resources, individuals, and even departments may move location, and have an affect on results. Furthermore, the behaviour of a component may change over time as information changes within the system. Actors can change behaviour as a result of handling a message.

[**modularity**] Each part of an organisation is intended to perform a business function that can be expressed in terms of a collection of operations. The internal organisation in terms of people, IT systems and the implementation of various business processes is usually hidden. The AMC provides an interface of message handlers for each actor. Both the state and the implementation of the message interface are hidden from the outside. The specification of an actor in terms of its external interface can be expressed in terms of LTL formulas that constitute the external goal for a component.

[**autonomy**] A key feature of an organisation is that the behaviour of each sub-component is autonomous. A particular department is responsible for its own behaviour and can generate output without the need for a stimulus. The AMC is highly concurrent with each actor being able to spawn multiple threads and over which other actors have no control (unless granted by the thread originator).

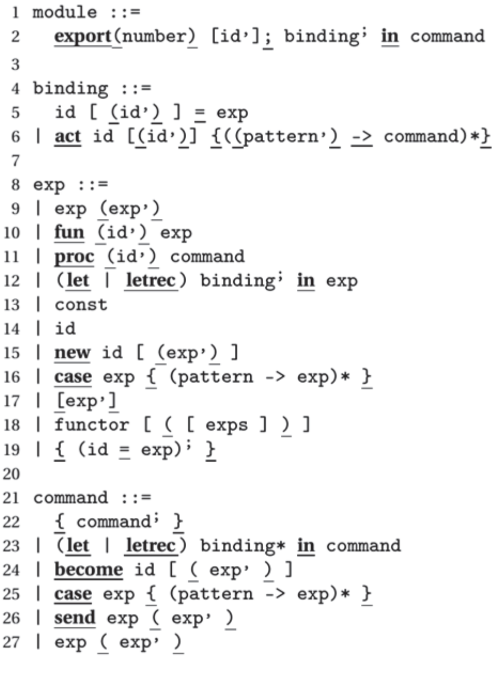
[**distribution**] An organisation may be distributed and this may be an important feature of its simulation. Furthermore, we have a requirement that the tooling for organisational analysis and simulation should support distributed concurrent development. The AMC is message-based and seamlessly supports execution in the same address space, via a network connection or in the cloud.

[**intent**] In addition to autonomous behaviour, an organisation component exhibits *intent*. This might take the form of an internal goal that guides the behaviour of the component to ensure that it contributes to the overall mission of the organisation. Although actors do not directly provides support for such goals, we intend to use results from the field of Multi-Agent Systems (van der Hoek, 2008) where support for goal-based reasoning is provided within each agent when determining how to handle messages.

[**composition**] An organisation is an assembly of components. As noted above, the topology of an organisation may be static or dynamic. Actors can be nested in more than one way. Actor behaviours are declared and new actors are dynamically created with an initial behaviour (much like Java classes). The scope of actor behaviours can be nested to provide modularity. Adding a dynamically created actor to the state of a parent actor provides composition. Such actors can be sent as part of messages. If the source actor retains the address, then the communicated actor becomes shared between the source and the target of the message.

[**extensibility**] Our aim is to support a number of simulation and analysis use-cases. As such the kernel language will need to support a collection of independent domains. Whilst we expect the DSLs to target the kernel language it is likely that each domain will have its own fundamental concepts and actions (so-called *Therbligs,* (Stanton, 2006)). We envisage such domain-specific features being defined in the kernel language and then pre-loaded to form an augmented target language for DSL translations.

[**event-driven**] Organisational components cannot rely on *when* communications occur and *where* they originate. In addition, a component may simply cause an event to occur without knowing who will consume the event. This is to be contrasted with message-based communication where the target is always known to the source and where sometimes the message carries information about the source that becomes available to the target. The AMC is based on message passing where the source knows the address of the target. Given that the kernel language is the target of DSL transformations, support for event-based communication becomes an architectural issue where events are simply messages that are sent to an actor container that is responsible for delivering event-messages to dynamically changing collections of actors. Providing that the transformation establishes the correct assembly of actors and conforms to an appropriate message passing protocol then component events are supported without needing to make them an intrinsic part of the kernel.



```
module ::=
  export(number) [id’]; binding; in command

binding ::=
  id [ (id’) ] = exp
| act id [(id’)] {((pattern’) -> command)*}

exp ::=
| exp (exp’)
| fun (id’) exp
| proc (id’) command
| (let | letrec) binding; in exp
| const
| id
| new id [ (exp’) ]
| case exp { (pattern -> exp)* }
| [exp’]
| functor [ ( [ exps ] ) ]
| { (id = exp); }

command ::=
  { command; }
| (let | letrec) binding* in command
| become id [ ( exp’ ) ]
| case exp { (pattern -> exp)* }
| send exp ( exp’ )
| exp ( exp’ )
```

Figure 4: The ESL Kernel.

Fig. 4 shows the kernel features of a language called ESL that is currently under development. It has been designed to support the features that are discussed in this section and, as a result, address the problems outlined in section 1. The syntax definition assumes const and id for constants and identifiers, underlines terminals and uses $\mathtt{x}^{\mathtt{y}}$ to denote `x(yx)*`. In overview, a system consists of a collection of modules (line 1) that exports a means of communication. An actor behaviour is introduced as a binding (line 6) that could be nested as a local definition (line 12). A behaviour has state and message handling rules (line 6) and a new actor is created (line 15) by supplying values for the state variables. Pattern matching is used to process messages (`pattern` is not defined) and to dispatch to a command. A command can change the behaviour of the receiver (line 24) or send further messages (line 26). Data values are constants (line 13), lists (line 17), terms (line 18), actors, functions (line 10) and procedures (line 11).

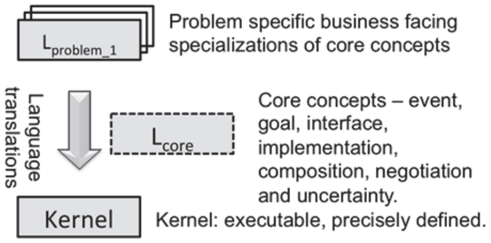

L_problem_1
Problem specific business facing specializations of core concepts
Language translations
L_core
Core concepts – event, goal, interface, implementation, composition, negotiation and uncertainty.
Kernel
Kernel: executable, precisely defined.

Figure 5: Translation to the Kernel.

Fig. 5 shows the proposed process where many different problem-oriented DSLs are translated in terms of the core concepts to the kernel language where simulation and analysis can be applied.

4 SUMMARY AND NEXT STEPS

We have identified 3 key problems and proposed an approach to solving these in order to make decision making in organisations more effective. Our approach is based on a domain analysis of the core concepts and involves supporting multiple DSLs that can be understood in terms of these concepts and realised in terms of a kernel language used for simulation and analysis. To date we have analysed several EA use-cases in terms of the core concepts and have started to prototype the kernel language. Currently we are developing real-world case-studies to validate and illustrate the proposed approach. For each case-study we will construct: a problem specific specialization of the core concepts, a business facing language constituting concrete syntax for the specialized meta model, and a mapping from this language to the kernel language.

REFERENCES

Tarr, P., Ossher, H., Harrison, W., and Sutton, S.(1999). N degrees of separation: multi-dimensional separation of concerns. Proceedings of the 21st Int. Conf. on Software Engineering, pp. 107-119.

Barros, T., Ameur-Boulifa, R., Cansado, A., Ludovic, H. and MadelaineBarros, E (2009). Behavioural models for distributed Fractal components. Annales des Télécommunications 64(1-2).

Bonabeau, Eric (2002). Agent-based modeling: Methods and techniques for simulating human systems. Proc. of National Academy of Science, USA, 99(Suppl 3).

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). Design patterns: elements of reusable object-oriented software. Addison Wesley.

Clark, A., and Barn, B (2011). Event driven architecture modelling and simulation. SOSE 2011.

Michelson Brenda (2006). Event-Driven Architecture Overview. Patricia Seybold Group, February 2, 2006.

Yu, E., Strohmaier, M. and Xiaoxue Deng (2006). Exploring Intentional Modeling and Analysis for Enterprise Architecture. Enterprise Distributed Object Computing Conference Workshops. EDOCW '06.

Clark, A., Frank, U., Kulkarni, V., Barn, B. and Turk, D (2013). Domain specific languages for the model driven organization. In Proc. of the 1st Workshop on the Globalization of Domain Specific Languages.

Zachman, J (1999). A framework for information systems architecture. IBM Systems Journal, vol. 38(2/3), 1999.

The Open Group, TOGAF 9.1 White Paper On Intro. to TOGAF Version 9.1 http://www.opengroup.org/togaf/

Wisnosky, D. and Vogel J. (2004). DoDAF Wizdom: A Practical Guide to Planning, Managing and Executing Projects to Build Enterprise Architectures Using the Department of Defense Architecture Framework (DoDAF).

Dardenne, A., Lamsweerde, A. and Fickas, S (1993). Goal-directed requirements acquisition. Science of Computer Programming, Volume 20(1–2).

Object Modeling Group, Business Motivation Model (BMM), v. 1.1, 2010, http://www.omg.org/spec/BMM/1.1/

Donella Meadows (2008). Thinking in systems: a primer. Chelsea Green Publishing.

Reinhartz-Berger, I., Cohen, S., Bettin, J., Clark, T., & Sturm, A. (2013) Domain Engineering: Product Lines, Languages and Conceptual Models. Springer.

Hewitt, C. (2010). Actor model of computation: scalable robust information systems. arXiv:1008.1459.

Hewitt, C. (2009). Norms and Commitment for iOrgs (TM) Information Systems: Direct Logic (TM) and Participatory Grounding Checking. arXiv:0906.2756.

van der Hoek, W., & Wooldridge, M. (2008) Multi-agent systems. *Handbook of Knowledge Representation*, 887-928.

Stanton, N. A. (2006) Hierarchical task analysis: Developments, applications, and extensions. *Applied ergonomics*, *37*(1), 55-79.

Business Requirements
Normative Approach to Behavior Modeling

Askhat Omarov, Rustem Kamun and Timur Umarov
Department of Computer Engineering, Kazakh-British Technical University, 59 Tole bi str., Almaty, Kazakhstan
{*askhat.omarov91, r.kamun*}*@gmail.com, t.umarov@kbtu.kz*

Keywords: Business Modeling, Organizational Semiotics, Deontic Logic, Responsibility, Requirements Formalization, Normative Positions, MEASUR.

Abstract: This paper attempts to formalize abstract requirements by amalgamating the MEASUR methodology with deontic logic and the limited version of the theory of normative positions. This amalgamation is performed in a manner that would potentially decrease the complexity inevitably embedded in the formalized abstract requirements so as to be able to target a wide range of non-technical users involved in the initial stages of software development lifecycle. This amalgamation is embraced by an artifact which represents a business model that avoids specifying software systems as a highly technical document and is rather referred to as a computational independent model of the MDA framework of software development.

1 INTRODUCTION

Business process engineering represents a critical area of concern in enterprise. It is becoming increasingly important to specify structured initial requirements for the business model which will (i) best fit the environment and the market demands and (ii) adapt to different changes that may happen. Since organizations are modeled using different theories, approaches, and solutions, it is our belief that, whichever approach is taken, the questions of complexity of modeling tools and insufficient depth of information description still remain.

Different new trends in requirements elicitation and gathering for the overall process of information systems' engineering were revealed. Among these approaches we can underline the one, which relates organizations to a set of interacting agents and entities while recognizing these objects via signs. This paradigm of modeling is regarded as organizational semiotics and represents an emergent discipline that studies the nature, characteristics, and features of information and how this information can be interacted between agents in the most optimal way within the contexts of organizations and business domains. OS finds its roots in semiotics and treats organizations as information systems, within which information is created, processed, distributed, stored and used.

The concept of organizational business modeling of the enterprise is driven by principles of adapting business processes to ever-changing market conditions in order to provide better services and products to clients. This is practically achieved by modeling individual business tasks and connecting them into logically organized activities to form workflows that would rapidly and effectively respond to the environment changes. Participants (humans or machines) are intimately related to these workflows by controlling and executing them in one way or another, subject to the knowledge possessed and the roles that these participants play. Equally important is the concept of contractual relationships formed among participants over the course of their interaction with an emphasis on social aspects of an organizational setting. However, most of the business processes with different functional ontological descriptions and annotations still lack complete operational semantics being defined within the requirements specifications.

In our attempt to address this problem, we relate to the following theories. One school of thought, led by Stamper and Liu, espouses a natural framework for the notions of behavioral patterns, roles, and relationships, conceptualized via the philosophy of language and communication, the theory of speech acts, and computational semiotics. Another school of thought, led by Kanger, Pörn, and Lindahl, supports so called the theory of normative positions which is a combination of deontic logic and logic of action and agency to provide a formal account of complex social interactions within organizations. As such concepts are new and not usually discussed in the wider fields of enterprise software engineering. Similarly, use of formal methods such as B and Event-B is a rare practice in the software engineering community for their inherent complexity in the software specification process. On the contrary, there also exist informal business process specification languages such as BPMN, BPEL, and WWF which are easier and handy in use.

There are several attempts to formalize requirements and provide formal semantics for unstructured or semi-formal requirements specifications. The authors in (Ponsard and Dieul, 2008) address this by bridging the semantic gap between the semi-formal requirements and formal models using a goal-oriented approach and regulation modeling and further refine them into formal specifications. Besides formalization an important role in developing, for example, critical systems is played by validation. This is thoroughly addressed by (Cimatti et al., 2010), in which the authors describe their methodology and a series of techniques for formalization and validation of high-level abstract requirements. The language used is a combination of first-order, temporal, and hybrid logic. Reliability as one of the most important properties of any systems, including critical ones, is raised by (van Lamsweerde, 2001) while discussing requirements engineering in terms of developing high-quality requirements. The arguments is built in the light of the related techniques for goal-oriented description of requirements, multiparadigm specification and so on.

This paper describes our approach and explains bearing of the modeling methodologies on information systems. We try to describe how the relatively informal methodology can incorporate normative techniques to make this language structured and computer interpretable for future machine use to implement model transformations. Because these concepts form a common root for modern logic and therefore formal methods, we find that we can potentially exploit our framework via model transformation ideas to derive a method for developing better business processes models from business requirements models. Our approach amalgamates the MEASUR methodology with the limited version of the theory of normative positions. This provides an additional level of semantics by “attaching” the notion of responsibility to the agents. The paper proceeds as follows:

- Section 2 provides an overview of the methodology MEASUR and gives a flavor of the normative approach in business modeling;
- Section 3 describes the proposed approach of formalizing MEASUR and defines the so-called normative tableaux;
- Section 4 provides an example of applying the approach in the example of making an electronic purchase and gives normative tableaux of several communication acts; and
- Section 5 briefly elaborates on future work of the research group.

2 BACKGROUND

In his definition of speech acts (Searle, 1969; Searle, 1971), Searle emphasizes the importance of so called proper circumstances in which speech acts have to be used. Uttering speech acts in proper circumstances causes construction or alteration of the world, e.g. creating an obligation. Searle also calls these proper circumstances rules of use. In our following discussion, term norms will be used to designate rules of using speech acts effectively.

2.1 Methodology MEASUR

Method for Eliciting, Analyzing and Specifying User Requirements (MEASUR) represents a radically new set of norm-oriented methods for business systems modeling and requirements specification for software development (Liu, 2000). MEASUR is a result of more than 30 years of research. The idea of MEASUR is that it sees the organizations as information systems with the set of agents, their corresponding set of potential actions (affordances) and set of norms which govern agents behavior. MEASUR offers five phases for business modeling and software development of which three the most important phases are problem articulation methods, semantic analysis method and norm analysis method.

The general ontological theory is concerned with fundamental questions of classifying everything that exists in the world into different categories, describing that everything while it exists, and seeking to elicit possible hierarchies and dependencies among the things that make up that everything. Liu (Liu, 2000) defines the term everything as constituent notions such as thing, entity, individual, universal, particular, substance, event, process and state. These notions are embraced with the general ontological study.

Although there are different types of ontologies related to knowledge and its representation, the only type of ontology that semantic analysis is relevant to is that which recognizes only our own behavior in accordance with our own ambiance. This type of ontology can be defined as a collection of representational data that models a certain domain of knowledge or discourse. Everything that exists in the world is dependent on the agents behavior. In other words, the meaning of a thing is related to the ability of the agent to recognize that thing as a particular entity.

Knowledge representation is an image of the world in our minds and is created by our perceptions of reality. The representation that we perceive, a surrogate, represents a substitute for the real entity itself. The correspondence between the entities and their surrogates forms the semantics for the representation. What we learn about the surrounding world forms our understanding about it. This understanding allows us to recognize the things that we know. Other things that we do not possess any knowledge about and hence cannot recognize them, do not exist in the sense of such entities in the world. By recognition we mean assigning meaning to things and knowing their properties. As we learn more about the world, the ontology changes accordingly. In other words, the possessed knowledge shapes a repertoire of our behavior.

With relation to the semantics, Liu (Liu, 2000) introduces the term affordance and defines it as a pattern of behavior of a human (or agent) which he learns through perceiving an ambient world. What an agent does is directed by his knowledge. The process of learning, or gaining knowledge about a certain set of facts, is in intimate relationship with the behavior and this relationship is bipartite. On the one hand, an agent learns the environment through his actions. On the other hand, the agents set of actions constrained by his behavior manifests knowledge that he possesses.

In any particular environment, whether an organization or a community, actions of its participants 1 must conform to the internal rules of behavior. Here the term behavior ensues from a social or a business context. Every agent has a set of certain actions to perform in accordance with the external conditions. The rules that members must adhere to is also referred to as norms. In the technical context, we are modeling the interaction between machines and humans, which can universally be referred to as agents. In terms of (Liu, 2000), the process of acting within the required norms represents invariant patterns of behavior or mechanisms of agents. Agents interact with each other which inevitably creates different relation patterns. The questions of modeling and formalizing these relations between agents is studied by the theory of normative positions.

The MEASUR methodology is a set of methods for business modeling and requirements specification for information systems. MEASUR can be used to analyze and specify an organization's business processes. As a traditional software system development lifecycle, MEASUR methodology of information systems modelling and requirements specification for software system development is divided into the following three stages (Liu, 2000):

1. Articulation of the problem, where a business requirements problem statement is developed in partnership with the client.
2. Semantic Analysis, where the requirements problem statement is encoded as an ontology, identifying the main roles, relationships and actions.
3. Norm Analysis, where the dynamics of the statement are identified as social norms, deontic statements of rights, responsibilities and obligations.

The processes of the first stage are comparable to other well known approaches to requirements specification. The last two stages require some elaboration and have their own unique associated notations.

The ultimate deliverable of these three stages is what we will refer to as a *normative ontology*. A normative ontology consists of what might be called a semi-formal requirements document, in the sense that it can be understood readily by clients, business analysts and developers. This model breaks down an information system into a set of business data, communicating agents (stakeholders, departments, computer programs) and business processes that agents can invoke in order to manipulate data between each other. The model consists of a role-and-relationships ontology together with a set of norms that formally define the structure and expected and permitted interactions within an organization and its processes.

2.2 The Normative Perspective

The theory of normative positions is based on the fundamental principles of deontic logic. Deontic logic is a formal system and represents a field of symbolic logic which is concerned with normative concepts such as obligation, permission, and prohibition used in contractual relationships for classifying actions and states of affairs. The word deontic comes from Greek and means "of that which is binding". Standard Deontic Logic (SDL, KD, or simply D) is the most studied and axiomatically defined system of deontic logic. SDL 1) represents a monadic deontic logic because its deontic operator is a one-place operator, 2) builds on propositional logic, and 3) is formally specified by a Kripke-style semantics. SDL uses the following notation to denote its main concepts: ObA stands for it is obligatory that A and PeA – it is permissible that A.

Deontic logic have not yet found wide application in technical areas such as Computer Science. However, deontic modalities could be efficiently used in specifying technical requirements for different business domains. For instance, modality obligatory may well model the requirement that a system must check whether a buyer have enough money in his back account before making a purchase in an on-line or physical store; and modality permissible can be applied to specify a rule that a bank may check (it is permissible that bank checks) credit history of a client before issuing a loan. More on the definition and a Kripke-style possible world semantics for SDL can be found in (McNamara, 2006).

The theory of normative positions was originally inspired by analytical study of law and originated in the works of Stig Kanger (Kanger, 1972; Kanger, 1985), Ingmar Pörn (Pörn, 1970; Pörn, 1977), and Lars Lindahl (Lindahl, 1977; Lindahl, 1994). The Kanger- Lindahl theory is characterized by attempts to apply modal logic – mainly standard deontic logic, the field of logic concerned with obligations and permissions – and the logic of action and agency to the concepts of legal and normative relations which Wesley Newcomb Hohfeld (1879–1918), an American jurist, had regarded to as the fundamental legal conceptions of jurisprudence (Hohfeld, 1964). These normative relations are alternatively referred to as normative positions that take the forms of obligations, permissions, duties, and rights of agents of a community, society or some other form of organization. By agents here we mean humans, machines, or both. The Kanger-Lindahl theory also embraces the formal representation of more complex normative positions such as entitlement, authorization and responsibility.

Besides the areas of legal knowledge representation (e.g. representation of laws, regulations, legal contracts, etc.), where the theory of normative positions found its initial application, there are also other areas such as Computer Science, where the theory contributed much for formal representation of relations between agents. For example, Jones and Sergot (Jones and Sergot, 1992; Jones and Sergot, 1993) describe a modified version of the Kanger-Lindahl theory and attempt to apply it to the problem of access control and security policies specifications and analysis for databases. In (Jones and Sergot, 1993), authors illustrate by an example of library regulations for governing the procedures of loaning books that the use of formal methods in developing system specifications have to be taken seriously whenever it is necessary to analyze an ideal case and an actual one and see how the actual behavior deviates from that of ideal. Use of such formal methods as deontic logic can help in revealing the possibility of violations, i.e. those deviations that had actually occurred. According to Jones and Sergot (Jones and Sergot, 1993), the use of deontic logic will in general allow (i) to reason with the specifications developed, (ii) to be able to test the internal consistency of the system specifications as a whole, and (iii) to use theorem provers to implement and test different components of the system.

In the works of Kanger, Pörn, and Lindahl (Kanger, 1972; Kanger, 1985; Pörn, 1970; Pörn, 1977; Lindahl, 1977; Lindahl, 1994), deontic logic was merged with logic of action and agency to provide a formal account of complex social interactions within organizations, which can be related to the technical context by applying it to the multi-agent environment. They introduce a so-called relativised modal operator which is designated as E$_a$, where a represents a responsible agent. The approach is partially similar to that of dynamic logic in the sense that it also assumes that an action, if performed, should bring about a certain state of affairs. For example, expression $[a]A$ would mean that after performing action a it is necessarily the case that A holds. In other words, a must bring about A. Analogously, $\langle a\rangle A$ means that after performing action a it is possibly the case that A holds, or a might bring about A.

However, in the theory of normative positions all actions are associated with their respective responsible agents which makes the semantics comparatively more expressive. When modeling business systems using this theory we now have to deal with the element of agent's responsibility embedded in the process of bringing about new states of affairs. The overall concept is formalized by

$$E_aA, \tag{1}$$

which is read as "agent a sees to it that A is the case" or similarly "agent a brings it about that A". It is important to note that actions in this case represent a relationship between agents and the state of affairs that these agents bring about.

The following expressions are properties for the action operator. The first axiom schema implies that the action operator is a *success* operator:

$$\vdash E_aA \rightarrow A. \tag{2}$$

It is read as: if agent a brings it about that A then A is indeed the case. The second property represents a rule of inference:

$$If \vdash A \leftrightarrow B\, then \vdash (E_aA \leftrightarrow E_aB). \tag{3}$$

Although the approach of combination of deontic logic with action logic is reminiscent of that of dynamic logic by its rules and operators, the theory of normative positions provides higher expressivity in a way that, unlike in dynamic logic: (i) by using operator E_aA one can express different atomic positions agent a can be in with respect to a particular state of affairs A; and (ii) using E_a operator gives another important advantage, namely, one can also formalise a normative interpersonal relationship by means of iterating the action operators:

$$E_aE_bA. \tag{4}$$

The examples of using this form of iteration can be demonstrated by the following:

$$E_a\neg E_aA \quad and \quad \neg PeE_b\neg E_aA \tag{5}$$

where the former effectively implies that agent *a refrains* from seeing to it that *A* and the latter means that the agent *b* is forbidden to prevent agent *a* from seeing to it that *A*. The detailed definition is described by Jones and Sergot in (Jones and Sergot, 1993).

The main building block of our proposed methodology for formalizing requirements is a tableau. It effectively uses the MEASUR constructs and the normative positions for describing the roles of the agents. A deeper analysis of the approach is required for clearer understanding of how MEASUR and the normative perspective are combined. We describe it in the next section.

3 FORMALIZING MEASUR

MEASUR and, in particular, NORMA, is a curiously mongrel beast. Its originator, Roland Stamper, while having a firmly applied background in system development, was influenced by ideas from philosophy of language and the expressive possibilities of deontic logic. However, in spite of drawing upon these ideas to develop the language and approach, it was always, first and foremost, a semiformal approach to requirements analysis, lacking a full formal semantics for analysis and reasoning.

The way in which MEASUR currently treats norms is analogous to the use of OCL within UML, that is syntaxes of OCL and UML are based on formal languages, but within their respective methods they do not have a precise semantics, which we give for MEASUR. In this paper, we provide what might be called a faithful formalization of MEASUR by restricting its norms to precisely the kind of logical language that inspired its notion of behavioral norm: that is, a limited version of deontic logic, combined with notions of agency inspired by the theory of normative positions. This will be defined as a first-order logic together with a straightforward semantics.

3.1 Normative Tableaux

The language proposed in this paper is somewhat more complex, due to its relational nature and conformance to an ontology. However, we need not study the full set of formulae given: we *restrict* our attention to a subset of normative formulae that correspond to the informal schema for behavioural norms given in Definition 3.1.

A MEASUR normative definition is of the form:

if *trigger* occurs and the *pre-condition is satisfied,*
then *agent* performs an action so that *post-condition*
is Obliged/Permitted/Impermissible from resulting.

Importantly, MEASUR norms never contain deontic quantifiers within the trigger, pre-condition or post-condition statements. Furthermore, actual communication acts are only mentioned in the post-condition: the trigger and pre-condition statement only refer to relations from the ontology. All three elements of the definition can refer to agents and entities, however. Finally, the prescription of an agent's responsibility and a given deontic obligation are only given in the implication of the constraint.

We can therefore restrict our attention to a subset of behavioral norms for a given ontology that naturally preserves these syntactic constraints within our formalization, now defined.

Definition 3.1 (Formal behavioral norms for an ontology)**.** *The set of* behavioral norms *is defined to be any formula of the form*

$$G \to E_a\mathbf{D}Post \tag{6}$$

where

- **D** *is a deontic operator Ob (obligatorily) or Pe (permissibly).*
- *The only free variables occurring in G and DEF are agent or entity variables from Var_{AGENT} and Var_{ENTITY}.*

The idea of a behavioral norm is to associate knowledge and information with agents, who produce and are responsible for it. From a philosophical perspective, truth is then defined as something that an agent brings about and is responsible for. From the perspective of determining how to implement a normative ontology as a workflow-based system, we view agents as corresponding to subsystems, business entities to specifications of data and behavioral norms to expected dynamic interaction protocols between subsystems.

MEASUR and, in particular, our logical restriction of MEASUR, allows much flexibility when detailing the intended meaning of communication acts. This can be done by clarifying assertions. When it comes to the question of implementation of a communication act, an analyst will always ask the client: what is entailed by this act? The client will then explain what changes the act is expected to make on the elements of the ontology. We encode this description as a *definition* DEF of the act A, of the form

$$A \to DEF \tag{7}$$

and

$$DEF \to A \tag{8}$$

henceforth abbreviated as

$$A \leftrightarrow DEF \tag{9}$$

where DEF is any MEASUR formula not involving communication acts, both A and DEF sharing the same free variables.

We are now ready to define our formal notion of a MEASUR requirement analysis document. Essentially, it consists of an ontology and pairs of behavioral norms and definitions, called normative tableaux.

Definition 3.2 (Normative tableaux)**.** *A requirements specification prescribes the action an agent is obliged (or permitted) to perform given the preconditions hold and consists of pairs of behavioural norms, each paired with definitions of the form*

$$REQ = \{(G_i \to E_{b\,i}\, D_i\, A_i, A_i \leftrightarrow DEF_i \mid i = 1,\ldots,n\} \tag{10}$$

where

- *each D_i is a deontic operator Ob or Pe.*
- *A_i is a single communication act.*
- *The only free variables occurring in G_i, DEF_i and A_i are agent or entity variables from Var_{AGENT} and Var_{ENTITY}.*

Each pair is called a normative tableaux.

A model $\mathcal{M}$ is a collection of these requirement pieces prescribed for some system and said to satisfy REQ if it validates the quantified conjunction of all REQ's normative tableaux. That is, $\mathcal{M}$ satisfies REQ when

$$\mathcal{M} \models \left(\begin{array}{l} \forall x_1 : T_1 \bullet \ldots, x_n : T_k \bullet \\ G_1 \to E_{b_1}\, D_1\, A_1 \wedge A_1 \leftrightarrow DEF_1 \\ \wedge \ldots \wedge \\ G_n \to E_{b_n}\, D_1\, A_n \wedge A_n \leftrightarrow DEF_n \end{array} \right) \tag{11}$$

where $x_1 : T_1, \ldots, x_n : T_k$ is the list of all free variables contained in the formulae of the tableaux.

A model for an ontology together with a set of behavioral norms $B_1, \ldots, B_n$ is one in which each norm is true for M. Depending on the model and the interpretation, this might be an abstract representation of a system execution, or might actually be an implementation of the specification: for example, one in which each possible world corresponds to an actual system state.

4 THE METHODOLOGY IN USE

This section will describe the methodology of specifying norms within the framework of new improved MEASUR. These norms specify, given certain conditions are true, an agent is obliged / permitted / impermitted to see to it that certain state of affairs is true. These types of normative specifications are not descriptive enough for requirements engineers to model changes in states of affairs with the necessary depth of data definition. Therefore, we need additional annotations for our norms, which we refer to as *definitions*. We hereafter present this norm/definition pair as so called *normative tableaux*.

In formalizing MEASUR, we illustrate norms as communication acts that are enabled by an agent, using the elements of agency and action of the theory of normative positions. In the framework of new improved MEASUR, in order to specify norms we use so called normative tableaux. Tableau is a table for defining norms and their data definitions. In other words, each normative tableau illustrates a norm as a pre-condition and a responsibility expression that relates an agent to a particular communication act taken from MEASUR ontology. A tableau also provides further notation for these norms in the form of definitions, so that each norm is paired with its definition. All norms have their definitions to obtain a certain depth of data definition and to be able to manipulate with these data. We need these definitions within our new MEASUR in order to provide ways for requirement engineers to define and describe data in conjunction with ontologies.

4.1 Case Study

The subject of our example is to process an order made by a customer using her credit card. The meaning of processing an order embraces herein such activities as receiving a new order, processing its data (price, customer's credit card information), invoicing and dispatching an order, interactions with warehouse and rejecting an order. In performing these activities in the order specified, we examine interaction between three involved players in an organized manner.

The first player, which starts the whole process of ordering is Customer. This player orders products, thereby creating a new instance of order and sends it to the electronic ordering system, which further processes it in a sequence specified below. The electronic ordering system eOrder initiates one activity after another and at some point starts interaction with yet another player called Warehouse. All these activities and communications between different players shift the order from one state to another. All orders have a strict correspondence with its particular customer and all orders are stored in the system during their processing until they are rejected. The activity of rejection removes the order from the system and cancels the correspondence with its particular customer.

The activities of making a purchase using our system is of the following order:

1. When a customer makes an on-line order, he initiates an interaction with the electronic ordering system by creating a new instance of the order and changing the status of this order to "received".
2. Electronic ordering system initiates the next activity which processes this new order by checking order's data and customer's solvency. If this check is successful, then the status of this order changes from "received" to "pending". Otherwise, if the order's total cost is higher than customer's credit card limit, then the status of his order changes to "rejected", which starts the rejecting process by removing the order from the system.
3. When the order's state changes to "pending" and if the product ordered is in stock, the system initiates the process of invoicing it by changing its status to "invoiced".
4. If the order is not available in the stock, then the electronic ordering system initiates a communication to the warehouse by starting the "request_increase" activity.
5. After successful invoicing, the order's status changes to "dispatched" to indicate that the order is dispatched.

The diagram for this example may be similar to UML's sequence or activity diagrams. But, it is different in the way of providing more expressive definitions for activities, interactions, and the notion of responsibility. We will employ this example to illustrate the use of the MEASUR methodology its formalized version.

We first represent knowledge as a domain ontology encompassing such entities as agents, actions, and relationships between them. The ontologies of MEASUR's semantic analysis are similar to those of, for example, OWL: they decompose a problem domain into roles and relationships. Ontologies enable us to identify the kinds of data that are of importance to business processes. A key difference with OWL is the ability to directly represent *agents* and *actions* as entities within an ontology. This is useful from the perspective of business process analysis, as it enables us to identify tasks of a workflow and relate them to data and identify what agent within the organization has responsibility for the task.

In our treatment, affordances are viewed as *classifications* of things within a business system, with an ontology defining a type structure for the system. An actual executing system consists of a collection of affordance *instances* which possess the structure prescribed by the ontology and obey any further constraints imposed by an associated set of norms.

In our case study, we model an electronic ordering system. The purchasing system player *eOrder* is related to the *Customer* agent and the *Warehouse* agent. *Customer* orders products from *eOrder* by initializing the interaction. *eOrder* receives data describing the order and performs further processing which normally includes checking availability of the product in the system, customer's solvency, invoicing, dispatching, and rejecting. If processing is successful, the system files an invoice for the purchase and subsequently dispatches it. If the product is unavailable, *eOrder* sends a request to the warehouse to increase the stock to further proceed with the order. If processing is unsuccessful, *eOrder* rejects the order.

An ontology for the purchasing system is given in Fig. 1. Agents are represented as ovals and business entities as rectangles with curved edges. Communication acts and relations are shown as rectangles, with the former differentiated by the use of an exclamation mark *!* before the act's name.

All affordances (including agents and business entities) have a number of typed attributes, defining the kinds of states it may be in. We permit navigation through an affordance's attributes and related affordances in the object-oriented style of the OCL.

The system involves processes that cross the boundaries of two subsystems: a customer agent, an order processing system, and a product warehouse system. These three subsystems are represented as agents in the ontology, *Customer*, *eOrder* and *Warehouse*, respectively. By default all agents except for *Customer* contain start and end attributes.

An order is associated with its customer, defined by the *ordered_by* relationship holding between the *customer* agent and *order* entity. An order can stand in an *ordered* relationship with the *eOrder* agent, after it has been successfully processed. Communication act *!receive_order* corresponds to the initial reception of data. The *Processing* communication act further deals with the newly arrived order and checks whether the client's credit limit allows for the purchase. Namely, it checks whether the total cost of the purchase is less than the credit limit of the customer. This condition results in the following outcomes: if the credit limit is lower than the total cost of the purchase then the system rejects the order, otherwise it initiates the invoicing process (denoted by

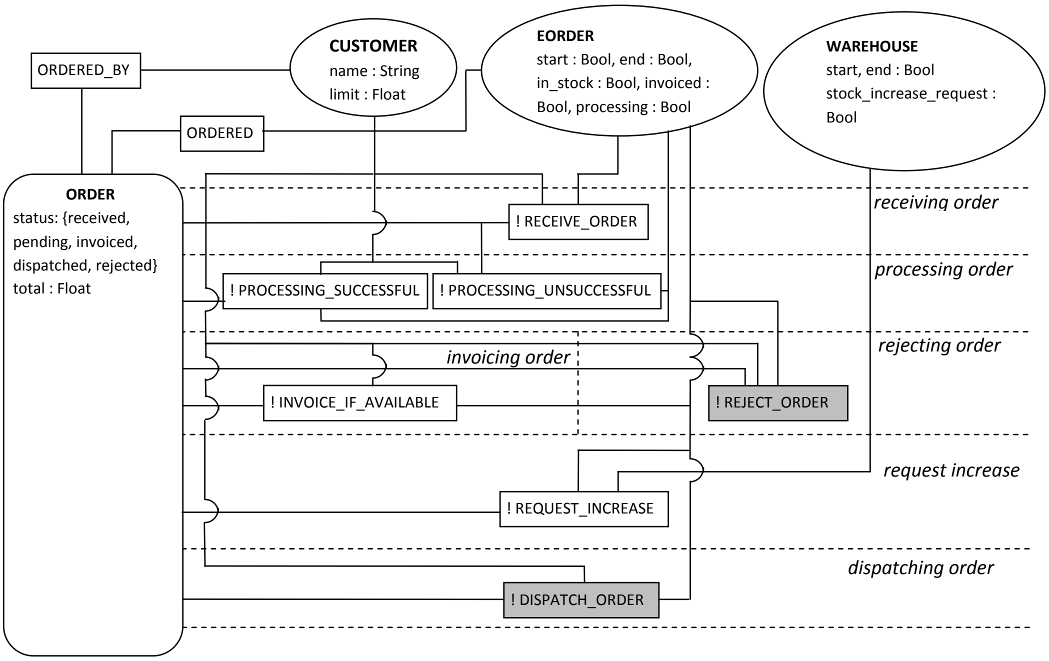


Figure 1: Example normative ontology.

the *invoice_if_available* communication act). It does so if the stock contains enough amount of the product for the order. If not, then the system requests to increase the stock by initiating *request_increase*, which sets flag *stock_increase_request* to true. Finally, the system dispatches the order by *dispatch_order*.

4.2 Behavioral Norms and Definitions

All norms described here strictly follow the norm-definition relationship (9) and take form according to (10). Consider the communication act *!receive_order* from our example, corresponding to the initial reception of data by the order processing system. The idea that this reception can only occur over orders that are not yet processed is captured by the normative tableaux shown in (12) and (13). Both relationships and communication acts are represented as logical relations in our language, but communication acts are not used in pre-conditions, and may only be placed *after* a Deontic operator.

Communication acts often define resulting changes of state on related agents and entities. As shown in our ontology in Fig. 1, *receive_order* relates three affordances: agents *Customer* and *eOrder* and business entity *Order*, instances of which are used as arguments for this communication act. As such, this communication act should affect the following relationships *ORDERED* and *ORDERED_BY* that are involved in relating the pertinent affordances. The meaning of the behavioral norms are extracted from additional definitions, which represent their equivalent meaning. An equivalent meaning of the reception of an order entails a change of state of affairs to include a newly arrived order, the status becomes set to “received”, and the system initiates the processing stage by setting its attribute to *true*. This is formalized by the norm

$$\begin{aligned}&\forall cc : Customer \bullet \forall oo : Order \bullet \forall e : eOrder \bullet\\&\quad \neg ordered\_by(oo,cc) \wedge \neg ordered(oo,e) \rightarrow\\&\qquad E_e\ Ob\ \mathbf{receive\_order}(\mathbf{oo},\mathbf{cc},\mathbf{e})\end{aligned} \tag{12}$$

and its definition

$$\begin{aligned}&\forall cc : Customer \bullet \forall oo : Order \bullet\\&\quad \forall e : eOrder \bullet \mathbf{receive\_order}(\mathbf{oo},\mathbf{cc},\mathbf{e}) \leftrightarrow\\&\qquad ordered\_by(oo,cc) \wedge ordered(oo,e) \wedge\\&\quad oo.status = received \wedge e.processing = \top\end{aligned} \tag{13}$$

The norm (12) and its definition (13) form a single normative tableau for the communication act *!receive_oder*. Note that we have employed the conventions and assumptions just given. So determinacy is implicit, and we are quantifying over individual formulae, to indicate types of variables. However, importantly, all variables – for example, *oo* and *cc* – should be understood as denoting the same possible interpreting object in the semantics.

To further proceed with the new order we define a norm for processing the data received. This norm effectively defines the requirement to check the payment (in order to be able to proceed with the order in a successful manner, it is important that the *total* for the purchase does not exceed the *limit* value in the credit card) and order related data. It should also be noted that another requirement to enable this norm is that the order should already be *received* from a *particular* customer. In order to model a system which behaves in a discrete fashion, norm *processing* is split into two cases: *process_successful*

$$\begin{aligned}&\forall cc : Customer \bullet \forall oo : Order \bullet \forall e : eOrder \bullet\\ &\quad ordered\_by(oo, cc) \wedge ordered(oo, e) \wedge\\ &\quad oo.status = received \wedge e.processing = \top \wedge\\ &\quad oo.total < cc.limit \rightarrow\\ &\quad E_e\ Ob\ \mathbf{process\_successful}(\mathbf{oo}, \mathbf{cc}, \mathbf{e})\end{aligned} \tag{14}$$

with its definition

$$\begin{aligned}&\forall cc : Customer \bullet \forall oo : Order \bullet \forall e : eOrder \bullet\\ &\quad \mathbf{process\_successful}(\mathbf{oo}, \mathbf{cc}, \mathbf{e}) \leftrightarrow\\ &\quad oo.status = pending\end{aligned} \tag{15}$$

and *process_unsuccessful*

$$\begin{aligned}&\forall cc : Customer \bullet \forall oo : Order \bullet \forall e : eOrder \bullet\\ &\quad ordered\_by(oo, cc) \wedge ordered(oo, e) \wedge\\ &\quad oo.status = received \wedge e.processing = \top \wedge\\ &\quad oo.total \geq cc.limit \rightarrow\\ &\quad E_e\ Ob\ \mathbf{process\_unsuccessful}(\mathbf{oo}, \mathbf{cc}, \mathbf{e})\end{aligned} \tag{16}$$

with its definition

$$\begin{aligned}&\forall cc : Customer \bullet \forall oo : Order \bullet \forall e : eOrder \bullet\\ &\quad \mathbf{process\_unsuccessful}(\mathbf{oo}, \mathbf{cc}, \mathbf{e}) \leftrightarrow\\ &\quad oo.status = rejected\end{aligned} \tag{17}$$

If the pre-conditions of *process_successful* hold then the norm changes the status to *pending*, as it is depicted in (14), which enables the *!invoice_if_available* communication act. Otherwise, the system initiates rejecting of the order by setting *status* to *rejected* as it is illustrated in (16). Below we show the norm and its definition for invoicing.

$$\begin{aligned}&\forall cc : Customer \bullet \forall oo : Order \bullet \forall e : eOrder \bullet\\ &\quad ordered\_by(oo, cc) \wedge ordered(oo, e) \wedge\\ &\quad oo.status = pending \wedge e.in\_stock = \top \rightarrow\\ &\quad E_e\ Ob\ \mathbf{invoice\_if\_available}(\mathbf{oo}, \mathbf{cc}, \mathbf{e})\end{aligned} \tag{18}$$

$$\begin{aligned}&\forall cc : Customer \bullet \forall oo : Order \bullet \forall e : eOrder \bullet\\ &\quad \mathbf{invoice\_if\_available}(\mathbf{oo}, \mathbf{cc}, \mathbf{e}) \leftrightarrow\\ &\quad oo.status = invoiced \wedge e.order\_invoiced = \top\end{aligned} \tag{19}$$

Invoicing the order and increasing the stock are followed by the dispatching process. We depict its definition.

$$\begin{aligned}&\forall cc : Customer \bullet \forall oo : Order \bullet \forall e : eOrder \bullet\\ &\quad ordered\_by(oo, cc) \wedge ordered(oo, e) \wedge\\ &\quad oo.status = invoiced \rightarrow\\ &\quad E_e\ Ob\ \mathbf{dispatch\_order}(\mathbf{oo}, \mathbf{cc}, \mathbf{e})\end{aligned} \tag{20}$$

$$\begin{aligned}&\forall cc : Customer \bullet \forall oo : Order \bullet \forall e : eOrder \bullet\\ &\quad \mathbf{dispatch\_order}(\mathbf{oo}, \mathbf{cc}, \mathbf{e}) \leftrightarrow\\ &\quad oo.status = dispatched\end{aligned} \tag{21}$$

Dispatching relates *Customer* to *eOrder* and *Order* and uses their instances for pre-conditions and to change the state of affairs towards updating the status to *dispatched*. This communication act finishes the ordering process. If processing was not successful, in other words if pre-conditions for norm *process_unsuccessful* were true, then the *eOrder* system rejects the order in accordance with the norm in (22).

$$\begin{aligned}&\forall cc : Customer \bullet \forall oo : Order \bullet \forall e : eOrder \bullet\\ &\quad ordered\_by(oo, cc) \wedge ordered(oo, e) \wedge\\ &\quad oo.status = rejected \rightarrow\\ &\quad E_e\ Ob\ \mathbf{reject\_order}(\mathbf{oo}, \mathbf{cc}, \mathbf{e})\end{aligned} \tag{22}$$

and its definition

$$\begin{aligned}&\forall oo : Order \bullet \forall e : eOrder \bullet\\ &\quad \mathbf{reject\_order}(\mathbf{oo}, \mathbf{cc}, \mathbf{e}) \leftrightarrow\\ &\quad \neg ordered(oo, e) \wedge \neg ordered\_by(oo, cc)\end{aligned} \tag{23}$$

According to Fig. 1, there exist other communication acts such as *invoice_if_available* and *dispatch_order* and their norm-definition pairs can also be defined in a similar way we have defined tableaux (12)–(23). The norm-definition relationship is important as expression in the definition part of the tableaux further elaborates each norm that precedes this definition. For example, the norm with its arguments *receive_order(oo,cc,e)* defined in (12) has a meaning of mapping certain objects together while defining a new instance of the order, in that:

- $ordered\_by(oo, cc)$ means that instance of order *oo* is mapped to the instance of an abstract machine *cc* (that models a customer),
- $ordered(oo, e)$ means that instance of order *oo* is mapped to the instance of machine *e* (that models a system),
- $oo.status = received$ means that the status should now change its value to *received*, and

- $e.processing = \top$ means that the processing status of the system should now change to *processing* to invoke the next norm.

All other subsequent norms are similar in the way they are read.

5 CONCLUSION

In this paper, we have shown an approach of combining the MEASUR methodology and the limited version of the theory of normative positions to model norms and normative ontologies. The level of abstraction pertinent to this approach effectively corresponds to early stages of business modeling. The model that we obtain as a result is a piece of knowledge which can (semi-formally) describe a functional structure of a company (e.g., stakeholders, departments, people, business processes, communications between these processes, etc.).

The approach described in this paper represents a combination of MEASUR with the theory of normative positions limited to the kinds of norms used in this methodology and describe business rules which specify interaction between two agents for achieving certain goals. Since the MEASUR methodology is a set of methods for business modeling and requirements specification for information systems, it can be used to analyze and specify an organization's business processes. The term affordance is introduced in this paper as a set of actions formed by the knowledge that an agent gains from existing in the environment and learning by interacting. The artifact developed can effectively be used as a high-level structured requirements specifications that can be turned into more concrete models using a range of different transformation and refinement techniques (e.g. Model-driven Architecture or refinement of B-based models), which will be the subject of our future work. We intend to see how the developed methodology can be useful in both developing formalized and structured requirements and semantically better fit the modular design specifications using formal languages such as B and Event-B.

REFERENCES

Cimatti, A., Roveri, M., Susi, A., and Tonetta, S. (2010). Formalization and Validation of Safety-Critical Requirements. *EPTCS*.

Hohfeld, W. (1964). *Fundamental Legal Conceptions As Applied in Judicial Reasoning*. Dartmouth Publishing.

Jones, A. and Sergot, M. (1992). Formal Specification of Security Requirements Using the Theory of Normative Positions. *Proceedings of the Second European Symposium on Research in Computer*.

Jones, A. and Sergot, M. (1993). On the Characterization of Law and Computer Systems: The Normative Systems Perspective. *Deontic Logic in Computer Science: Normative System Specification*.

Kanger, S. (1972). *Law and Logic*. Theoria.

Kanger, S. (1985). On Realization of Human Rights. *Action, Logic and Social Theory*.

Lindahl, L. (1977). *Position and Change – A Study in Law and Logic*. D. Reidel Publishing Company.

Lindahl, L. (1994). Stig Kanger's Theory of Rights. *Logic, Methodology and Philosophy of Science*.

Liu, K. (2000). *Semiotics in Information Systems Engineering*. Cambridge University Press.

McNamara, P. (2006). Deontic Logic. *Stanford Encyclopedia of Philosophy*.

Ponsard, C. and Dieul, E. (2008). From Requirements Models to Formal Specifications in B. *ReMo2V*.

Pörn, I. (1970). *The Power of Logic*. Blackwell Oxford.

Pörn, I. (1977). *Action Theory and Social Science: Some Formal Models*. D. Reidel, Dordrecht.

Searle, J. (1969). *Speech acts: An Essay in the Philosophy of Language*. Cambridge University Press.

Searle, J. (1971). *The Philosophy of Language*. Oxford University Press.

van Lamsweerde, A. (2001). Building formal requirements models for reliable software. *Reliable Software Technologies*.

Towards an Integrated Model for Enterprise Interoperability

Wided Guédria
SSI, CRP Henri Tudor, 29 JFK, Luxembourg-Kircheberg, Luxembourg
wided.guedria@tudor.lu

Keywords: Enterprise Interoperability, Interoperability Maturity, Framework, Integrated Model.

Abstract: One of the challenges faced by a network of enterprises is the development of interoperability between its members. Transformations in this context are usually driven by Enterprise Interoperability (EI) problems that may be faced. In order to quickly overcome these problems, enterprises need characterizing and assessing interoperability to be prepared to establish means for collaboration. Maturity models have been developed in response to this challenge. In this paper, we propose to define an integrated model based on a maturity model and an ontological formalization of the enterprise interoperability domain. This will allow diagnosing interoperability problems when assessing interoperability. The integrated model could be used improve the capability of an enterprise to interoperate based on a shared understanding of interoperability.

1 INTRODUCTION

As information systems and technologies grow in complexity and scope, the need for a coherent and comprehensive modeling approach becomes of paramount importance. The enterprise architecture deals with these issues in a coherent and integral fashion while at the same time a medium to achieve a shared understanding and conceptualization among all stakeholders involved and govern enterprise development based on this conceptualization (Proper, 2008). The intent of an enterprise architecture is to determine how an organization can most effectively achieve its current and future objectives. Among these objectives, we focus on improving the capability of the enterprise to interoperate with future partners.

The complexity of an organization can be huge due to many processes, departments and information systems involved. To provide architects with some structure, architecture frameworks have been introduced. These frameworks intend to aid architects by providing a meta-model (e.g. an ontology) which uses different abstraction levels to map all kinds of information needed. As part of EA, some of the models should be dedicated to interoperability, as an important issue that have to be taken into account to reach the business objectives. To our knowledge, none of the existing Enterprise Interoperability (EI) frameworks proposes a shared understanding and conceptualization of interoperability while improving the capability of an enterprise to interoperate.

In this paper, we propose to define an integrated framework that defines and improves the enterprise interoperability. The paper is structured as follows: section 2 defines the research ontext and reviews some of the related work. Based on that, section 3 defines and describes the integrated framework. Section 4 gives an example to use the proposed model. Conclusion and future work are outlined in section 5.

2 RELATED WORK

In the current globalised and networked society, enterprises need to collaborate with other enterprises to meet their own added values and to exploit the market opportunities. A major issue in global collaboration and cooperation is the development of interoperability. Many definitions of interoperability have been proposed in the literature. A review of them can be found in (Guédria, 2012). The most known definition is the one proposed by the IEEE, considering interoperability as the ability of two or more systems or components to exchange information and to use the information that has been exchanged (IEEE, 1990).

2.1 Enterprise Interoperability Frameworks

The main purpose of an interoperability framework is to provide an organising mechanism so that concepts, problems and knowledge on EI can be represented in a more structured way (Chen et al., 2008). So far, the most known EI frameworks are: ATHENA (Advanced Technologies for interoperability Heterogeneous Enterprise Networks and Applications) Interoperability Framework (AIF) (Ruggaber, 2006), the European Interoperability Framework (CompTIA, 2004), the E-health interoperability framework (NEHTA, 2005) and the Framework for Enterprise Interoperability (FEI). The review of the different aspects and the frameworks coverage of these frameworks lead to identify the main elements in EI context (i.e. (a) the three interoperability aspects: conceptual, technical and organisational, (b) four concerns of EI: business, process, service and data); however, none of the EI frameworks defines interoperability, or proposes to improve it.

2.2 Ontology of Enterprise Interoperability (OoEI)

The OoEI aims at formally defining Enterprise Interoperability (EI) while providing a framework to describe problems and related solutions pertaining to the interoperability domain. Figure 1 gives an extract of the OoEI meta-model. Interoperability concerns, define the content of interoperation that may take place at various levels of the enterprise (data, service, process, business). Interoperability barriers identify various obstacles to interoperability in three categories (conceptual, technological, and organizational).More details can be found in (Guédria, 2012).

2.3 Maturity Model for Enterprise Interoperability (MMEI)

A maturity model is a framework that describes for a specific area of interest a number of levels of sophistication at which activities in this area can be carried out (Alonso et al., 2010). In our case, the specific area of interest is EI. EI maturity can be measured in two ways: *a priori* where the measure relates to the potentiality of a system to be interoperable with a possible future partner whose identity is not known at the moment of evaluation, *a posteriori* where the measure relates to the compatibility measure between two (or more) known systems willing to interoperate. Developing interoperability can induce many problems that have to be solved in order to achieve targeted objectives. Solving these problems may be a long iterative procedure which can fail due to the lack of a consensus between partners or the high cost of the solution applicability. Preventing and solving interoperability problems before they occur is simpler and usually less costly than developing corrective actions. Within this context, *a priori* assessment deserves particular attention in order to help enterprises knowing their strengths and weaknesses in terms of interoperability and undertaking improvement actions. Many maturity models have been developed in the literature. Among them, we find the Maturity Model for Enterprise Interoperability (MMEI) which is the only one defined within an *a priori* context of interoperability. A review and a comparison of them could be found in (Ford, 2008), (Guédria et al., 2008), (Guédria, 2012).

MMEI (Guédria et al., 2013) allows companies to evaluate their potentiality to interoperate, in order to know the probability that they have to support efficient interoperation and to detect precisely the weaknesses that are sources of interoperability problems. Moreover, MMEI differs from all other maturity models dedicated to interoperability so far. It is intended to cover the three interoperability levels (conceptual, technological, and organizational) at each of the EI concerns (business, process, service, data), as defined in the Framework for Enterprise Interoperability (FEI) (Chen, 2006). MMEI defines five levels of interoperability maturity (Guédria et al., 2013). A general view of the MMEI model with its contents is given by Table 1. Each one of the maturity levels is an instantiation of this general view with an evolution of the content regarding the evolution of the level.

3 INTEGRATED MODEL

In this section, we propose to define an integrated model allowing improving EI while sharing a common understanding of it. This integrated model comes in response to a research gap that we have been identified when reviewing the interoperability related work in the previous section. Indeed, each one of the presented interoperability frameworks and models has a unique purpose as presented in Figure 2.

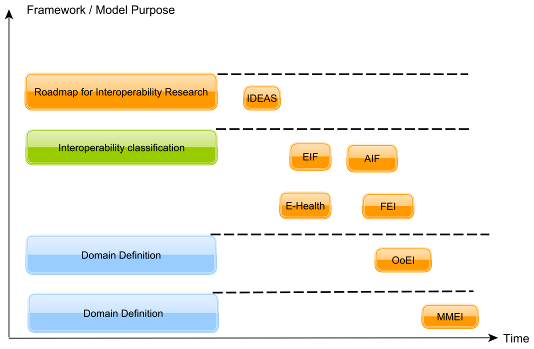

Framework / Model Purpose
Roadmap for Interoperability Research
IDEAS
Interoperability classification
EIF
AIF
E-Health
FEI
Domain Definition
OoEI
Domain Definition
MMEI
Time

Figure 2: Enterprise Interoperability Frameworks and Purposes.

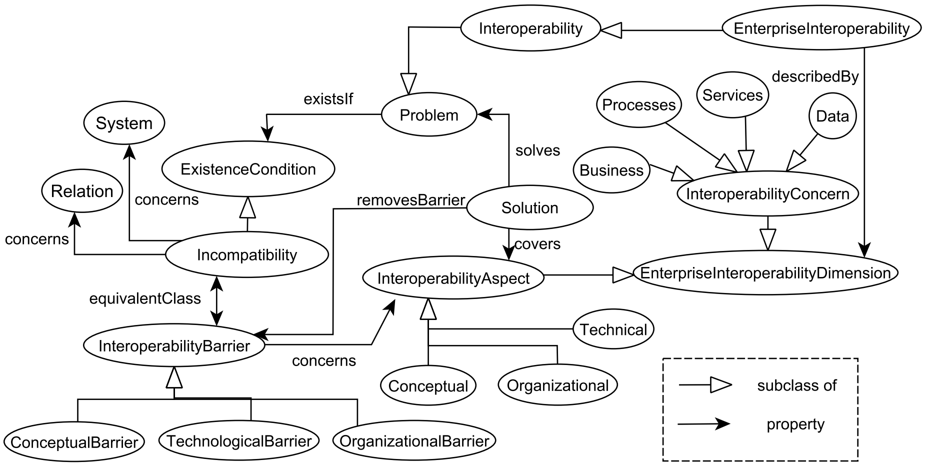


Figure 1: Extract of the OoEI.

Table 1: General view of MMEI levels structure.

	Conceptual	Technological	Organizational
Business	Business models, enterprise visions, strategies, objectives, policies	Infrastructure, technology	Work methods, business rules, and organizational structure
Process	Processes models	Tools supporting processes modeling and execution	Responsibilities, Process management and rules
Service	Services models	Tools supporting services and applications	Responsibilities, service and application management and rules
Data	Data models, (semantic, syntax)	Data storage and exchange devices	Responsibilities, data management and rules

To deal with this gap, this section proposes to define the integrated model based on the OoEI and the MMEI models.

The integration of the MMEI and OoEI will allow the enrichment of the OoEI with concepts and details from the MMEI, allowing the enterprise to better diagnose its interoperability problems and find suitable solutions. On the other hand the OoEI contains a considerable knowledge that may help the MMEI assessing the Interoperability readiness of a given enterprise.

The structure of MMEI is based on a simplified version of FEI where we can find the EI concerns (i.e. Business, process, service and data) and interoperability barriers (i.e. conceptual, technological and organizational), as depicted by table 1. These concepts can also be found in the OoEI (Naudet et al., 2008).

Moreover MMEI and OoEI follow a systemic view (Von Bertalanffy, 1968), where an enterprise is seen as a complex system. The content of each cell of the table 1 can be related to the OoEI. Figure 3 shows how these cells can be related to the OoEI. The OoEI concepts are presented with white ellipses while concepts related to the MMEI model are presented with gray color.

Based on the OoEI and MMEI review, three main dimensions of EI are considered: Interoperability aspects (conceptual, organizational and technical), EI concerns (business, process, service and data) and Interoperability barriers (Conceptual, Organizational, and Technological). These are represented by the concepts: InteroperabilityAspect, InteroperabilityConcern and InteroperabilityBarrier respectively. These are all modeled with their different constituents represented here as dimensions describing Enterprise Interoperability, as shown in figure 3.

Within the context of EI, interoperability problems are represented by the InteroperabilityBarrier concept. The term barrier is defined as an incompatibility, obstructing the sharing of information and preventing exchanging services (Chen et al., 2006).

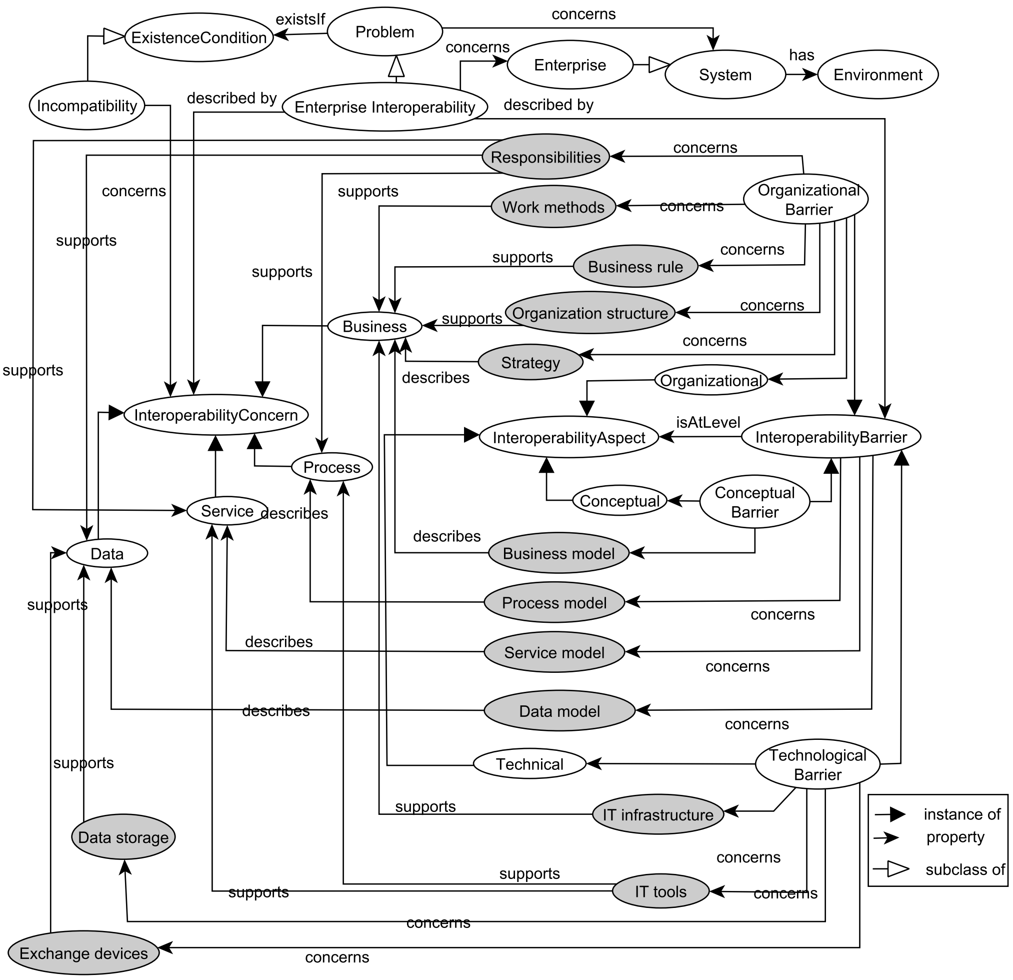


Figure 3: Extract of the integrated model.

The establishment of interoperability (with its three aspects) consists of removing identified barriers (conceptual barrier, organizational barrier or/and technological barrier). Hence each InteroperabilityBarrier is related to the corresponding InteroperabilityAspect (see figure 3).

For each OrgaizationalBarrier, the criteria that need to be verified are the definition and compatibility of the Responsibilities, Work methods, Business rules, Organization structure and strategy, as defined by the MMEI model (see table 1). This is represented by the concepts Responsibilities, Work methods, Business rules, Organization structure and strategy which are considered by the Organizational Barrier. Similarly, the concepts of Business model, Process model, Service model and Data model are concerned with the ConceptualBarrier and the concepts of IT infrastructure, IT tools supporting data storage and Exchange devices are concerned with the TechnologicalBarrier.

This model can be used thereafter to have the required knowledge to assess the EI of the considered enterprise. The gray MMEI concepts (gray ellipses) in the figure 3 presents the requirements and related information that need to be verified. For example, in order to assess the EI at organizational level, some of the requirements that assessors have to verify are whether responsibilities supporting business, process, service and data interoperability concerns are properly defined and that are compatible with those used within the enterprise environment.

4 ILLUSTRATIVE EXAMPLE

To better understand the application and the use of the above defined conceptual framework, it is interesting to check its utility by an illustrative example. A potential interoperability problem exists if the considered enterprise, say E has no defined models, tools, rules, etc. or that it uses incompatible ones with those used in its environment (including its partners, clients, providers, etc.). For example, if E adopts 50 hours of employment per week while being situated in France or Luxembourg, this will create incompatibility with the environment of the enterprise where a maximum of 40 hours is fixed for the employment. Figure 4 illustrates this interoperability problem that can be detected when assessing the interoperability of E towards unknown partner.

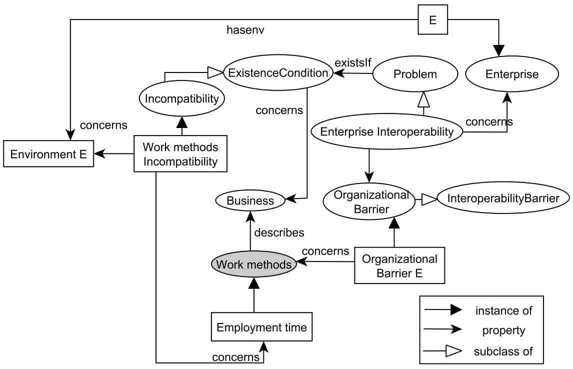


Figure 4: Illustrative Example with the enterprise E.

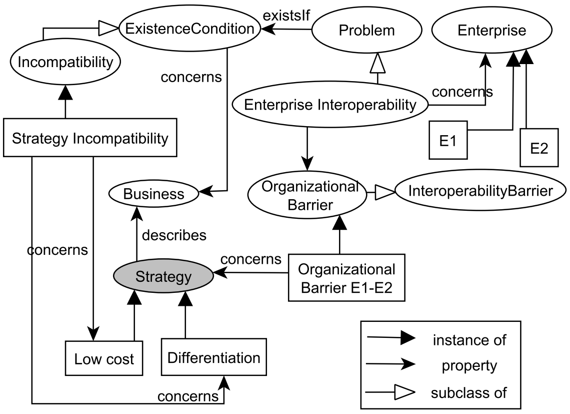


Figure 5: Illustrative Example with enterprises E1 and E2.

Suppose now that a particular interoperability project starts (i.e. the partner is known but the interoperations haven't started yet), barriers to interoperability can exist at each level of the company and of its partner. An interoperability problem exists if the two companies have different strategies. For example, one of the company, say E1 adopts a Low cost strategy which focuses on producing the product by lowering costs and maximizing income (by increasing volume), while E2 is adopting a differentiation strategy which focuses on producing different products (better) than those of competitors. Differentiation usually includes the processes for taking your product directly to consumers. So processing and marketing are involved. Differentiation in small markets often involves creating a personal relationship with consumers. An application of the defined conceptual model is presented in figure 5.

5 CONCLUSIONS

In this paper, we have proposed an integrated model based on the Maturity Model of Enterprise Interoperability (MMEI) and the ontology of Enterprise Interoperability (OoEI) concepts. The integration was facilitated by the common basis of the two models (systemics) and the use of the Framework of Enterprise Interoperability (FEI) by both of them. The integrated model allows having the required knowledge for the interoperability assessment and save human efforts in gathering information and validating it for each assessment. The interoperability assessment based on the integrated model allows diagnosing interoperability problems. This is facilitated by the nature of the OoEI which were conceived in a problem solving perspective. Future work are planned to improve this first version of the integrated model in order to implement an automatic assessment tool. The idea is to develop an ontology-based software infrastructure for web-based self-evaluation for enterprise interoperability. This tool will help to diagnose interoperability problems. This can be complemented by proposing solutions to interoperability problems, using best practices that have been developed for the MMEI (Guédria, 2012).

REFERENCES

Alonso, J., de Soria, I. M., Orue-Echevarria, L., and Vergara, M. (2010). Enterprise collaboration maturity model (ecmm): preliminary definition and future challenges. In *Enterprise Interoperability IV*, pages 429–438. Springer.

Chen, D. (2006). Enterprise interoperability framework. In *EMOI-INTEROP*.

Chen, D., Daclin, N., et al. (2006). Framework for enterprise interoperability. In *Proc. of IFAC Workshop EI2N*, pages 77–88.

Chen, D., Doumeingts, G., and Vernadat, F. (2008). Architectures for enterprise integration and interoperability: Past, present and future. *Computers in industry*, 59(7):647–659.

CompTIA (2004). European interoperability framework - ict industry recommendations. White paper, CompTIA, Brussels.

Ford, T. (2008). Measuring System Interoperability: An i-Score Improvement. *Proceedings of the 6th Annual Conference on Systems Engineering Research*.

Guédria, W. (Bordeaux, France, 2012). A Contribution to Enterprise Interoperability Maturity Assessment. Ph.D. thesis, University of Bordeaux1.

Guédria, W., Naudet, Y., and Chen, D. (2008). Interoperability maturity models–survey and comparison–. In *On the move to meaningful Internet systems: OTM 2008 Workshops*, pages 273–282. Springer.

Guédria, W., Naudet, Y., and Chen, D. (2013). Maturity model for enterprise interoperability. *Enterprise Information Systems*, (ahead-of-print):1–28.

IEEE (1990). *IEEE standard computer dictionary: A compilation of ieee standard computer glossaries*. Institute of Electrical and Electronics Engineers.

Naudet, Y., Latour, T., and D.Chen (2008). A systemic approach to interoperability formalization. In *IFAC08 workshops*.

NEHTA (2005). Towards an interoperability framework, version 1.8. White paper, National E-Health Transition Authority, Australia.

Proper, E. (2008). *Enterprise Architecture: Creating Value by Informed Governance*. Springer.

Ruggaber, R. (2006). Athena-advanced technologies for interoperability of heterogeneous enterprise networks and their applications. *Interoperability of enterprise software and applications*, pages 459–460.

Von Bertalanffy, L. (1968). *General System Theory: Foundations, Development, Applications*. Georges Braziller, Inc., New York, USA.

Applying Business Process Modeling Tools in Enterprise Resource Planning System Replacements
A Case Study

Stephan Groß[1] and Justus Holler[2]

[1]*Prof. Becker GmbH, Luetke Berg 4-6, 48341 Altenberge, Germany*

[2]*University of Muenster – ERCIS, Leonardo-Campus 3, 48149 Muenster, Germany*

stephan.gross@prof-becker.de, justus.holler@ercis.uni-muenster.de

Keywords: Change Management, Business Process Modeling Tool, Enterprise Resource Planning.

Abstract: Enterprise resource planning systems are the IT backbone of companies in the IT markets and in all those where efficient and integrated business processes are a necessity to stay competitive. As the markets change also the IT requirements do and the need for an Enterprise resource planning system replacement rises. This article argues that business process modeling tools can be used to document and communicate changes in the operational and organizational structure related to the replacement. The suitability of a tool with typical modeling features is evaluated within a project. The findings regarding semantics, structure, documentation, and re-usability are discussed and an outlook for future research is given.

1 MOTIVATION

In order to meet the constantly rising market expectations information technology (IT) does matter (Carr, 2003). Especially for large companies with complex and interleaved processes a well-designed Enterprise resource planning system (ERP system) supporting all business processes (Davenport, 1998) is of particular importance. During ERP system usage a point will be reached where the individual adaptation (in case of individual software) or the updates (in case of a standard ERP system e. g. SAP AG's ERP Central Component, SAP ECC) are not satisfying the customer needs anymore or are too costly. At this point in time the replacement of the complete ERP system is required. The migration from the old system to the new one is a complex and time-consuming project (Huang, Hung, Chen & Ku, 2004; Peslak, Subramanian & Clayton, 2008). This project and its subprojects are nearly always supported by external consultants because the required ERP system knowledge is not present and the human resources are missing.

In such projects tools like Microsoft Project or more sophisticated tools are often used. In this article we focus on business process modeling tools (BPM tools) supporting the change management subproject. Literature states, that these tools have a positive influence on critical success factors like understanding the current and new business processes including their IT and organizational interrelations (as-is and to-be models) (Holland & Light, 1999; Nah, Lau & Kuang, 2001). Hence, we argue that BPM tools are particularly well suited in documenting and communicating the changes.

But often the usage of the BPM tool is a project in its own. A full grown tool like the Architecture of Integrated Information Systems Toolset (ARIS Toolset) (Scheer & Nüttgens, 2000) is too complex for simply supporting the external consultants with their documentation. In contrast, tools like Microsoft Visio or Microsoft PowerPoint are too limited regarding their modeling functionalities. Therefore, the research questions of this article are:

- RQ1: *Are BPM tools well suited for documenting and communicating changes related to ERP replacements?*
- RQ2: *Which features are especially relevant for a BPM tool in order to efficiently support the documentation and communication of changes in the context of an ERP replacement?*

The remaining article is structured as follows: in the next section the case and research method is presented. In Section three the main features of the BPM tool within the running project are discussed. The last section concludes the findings and outlines future research potential.

2 RESEARCH METHOD IN THE PROJECT

One of the main discounters operating across Europe is replacing its individual piece of ERP software by SAP ECC for Retail. This happens because the old software did not meet the clients' expectations anymore regarding integrated processes, centralized data management, and performance. An agile method for ERP replacement is chosen in order to have a runnable prototype at an early stage in the project. The prototype is developed continuously with all involved stakeholders. The main characteristics of the project and company are depicted in Table 1.

Table 1: Project and company overview.

Dimension	Description
Purpose	ERP replacement
Method	Agile
Duration	Summer 2012 – approx. summer 2014
Domain	International Warehousing
Articles	Food and Non-Food
Employees	Approx. 80 (only employees involved)
Customers	Business-to-business customers
Budget	Approx. 3,000,000 euro

The project has several subprojects like migration, test management, support, training, and change management. Within the latter subproject one goal is to determine and manage the changes related to the system replacement. This is done in the three phases identification, documentation and communication as depicted in Figure 1 and described in the following subsections.

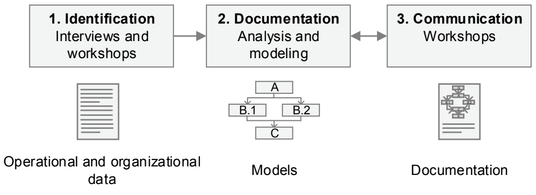


Figure 1: Research steps.

2.1 Identification Phase

In the first phase of the research 9 interviews are conducted in order to learn about the as-is processes. The external consultants also have the opportunity to observe the daily business and to take notes. The results of this phase are the as-is processes including descriptions regarding IT systems and responsibilities. Furthermore, the interviewees are asked for possible improvements from their point of view. This feedback is incorporated in the to-be processes directly. It is also explicitly addressed in the communication phase in order to improve the acceptance of the new to-be processes.

2.2 Documentation Phase

On the one hand the changes were anticipated within the business processes (gap analysis between as-is and to-be processes) and on the other hand within the organizational structure. The findings of the analysis and the input of the interviewees and workshops help to optimize the business processes, determine staff requirements (new positions or changed position descriptions within the organizational hierarchy) and support the key- and end-user trainings for the new ERP system (communication phase).

Due to constraints in the project there is no time for implementing a project specific tool supporting the process analysis. Hence, a BPM tool is selected which is suitable for the purpose of describing the operational and organizational structure changes. The selection of the tool is based on the following key features:

- Business processes modeling on multiple levels
- Support of organizational and IT structures
- Central glossary for project terms and model elements
- Modeling conventions enforced
- Re-usable (reference) models
- Easy access for stakeholder

After a tool (Becker, Clever, Holler & Shitkova, 2013) satisfying all the above features is selected the collected data in phase 1 is documented with the tool. The findings regarding the suitability of such a BPM tool in the context of an ERP replacement is discussed in section 3.

2.3 Communication Phase

In the last phase the outcome is presented and discussed in workshops and the process models are adjusted were necessary. Also the final documentation is created including all organizational, operational and IT system changes related to the ERP replacement. The features the BPM tool provided regarding communication are discussed in the next section, e. g. stakeholder tool access or reporting and export functionalities.

3 KEY FEATURES OF A BUSINESS PROCESS MODELING TOOL WITHIN SYSTEM REPLACEMENTS

The following section discusses the suitability of the above mentioned and further features (Table 2) for supporting the change management within an ERP replacement clustered in the following sub-sections.

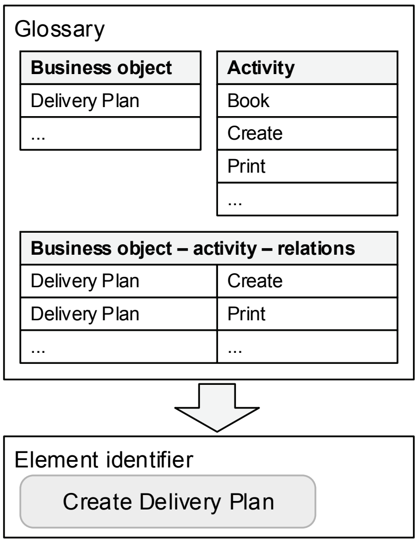
Table 2: Key features in the project.

Feature	Cluster
Syntax checker	Semantic standardization
Integration of semantics	Semantic standardization
Inclusion of modeling conventions	Semantic standardization
Integration of a glossary	Semantic standardization
Harmonized layers of abstraction in processes	Structured models
Organization charts	Structured models
IT system charts	Structured models
Attribution	Integrated documentation
Export and reporting	Integrated documentation
Reference process elements	Re-use of knowledge
Best-practice models	Re-use of knowledge

3.1 Supporting Model Quality by Semantic Standardization

An often mentioned tool feature is the possibility to support the user in checking the model syntax. This is definitely a nice feature to enhance the model quality. In this context it is questionable if and how model quality can be measured by means of syntactical correctness. Although many metrics are defined in literature for model quality (Moody, 2005) the most important quality aspect of a syntactically and semantically correct model is its "fitness for use" with respect to its purpose (Becker, Rosemann & von Uthmann, 2000; Rittgen, 2009). A model as abstract representation of reality should exclude not relevant details and focus on the main aspects regarding the purpose of the model (Dean, Orwig, Lee & Vogel, 1994). Typical purposes of process models are to cope with the process complexity, to document it, and to support conceptualization, analysis and communication (Dean, Orwig & Vogel, 2000; Ould, 1995; Van Hee, 1994; Becker, Kugeler & Rosemann, 2011). For satisfying these purposes a process model has to take the semantics of the process into account. Hence, the tool should foster semantic correctness by increasing the clarity, comparability, readability, and manageability of the models by implicit inclusion of modeling conventions (Mendling, Reijers & van der Aalst, 2008; Becker, Rosemann & Schütte, 1995; Becker, Clever, Holler, Püster & Shitkova, 2013a).

A further aspect of semantic standardization is the usage of a glossary. All business objects, activities, and allowed relations between them are defined within it. In the best case only the content of the glossary is (re-)used for describing the process element identifier by using activity-object phrase structures. For example, there is a glossary in which the business object "delivery plan" and the activity "create" are included. In addition, the business object "delivery plan" is associated with the activity "create". Hence, the element identifier "create delivery plan" is valid based on this glossary (Figure 2). By this means, uniform terms are used and misleading synonyms and homonyms are avoided (Becker, Richter & El-Hawari, 2010; Becker, Clever, Holler, Püster & Shitkova, 2013b; Breuker, Pfeiffer & Becker, 2009).



Glossary

Business object
Delivery Plan
...

Activity
Book
Create
Print
...

Business object – activity – relations	
Delivery Plan	Create
Delivery Plan	Print
...	...

Element identifier

Figure 2: Glossary and element identifier.

In terms of change management the glossary is particularly helpful because it contains all terms used within the as-is and to-be processes in one central index. It also includes definitions of new ERP specific terms, e. g. the SAP info record as master data for the connection between suppliers and articles. By this means the tool can help to standardize the used vocabulary within the models and therefore in the project.

3.2 Clear Structured Models

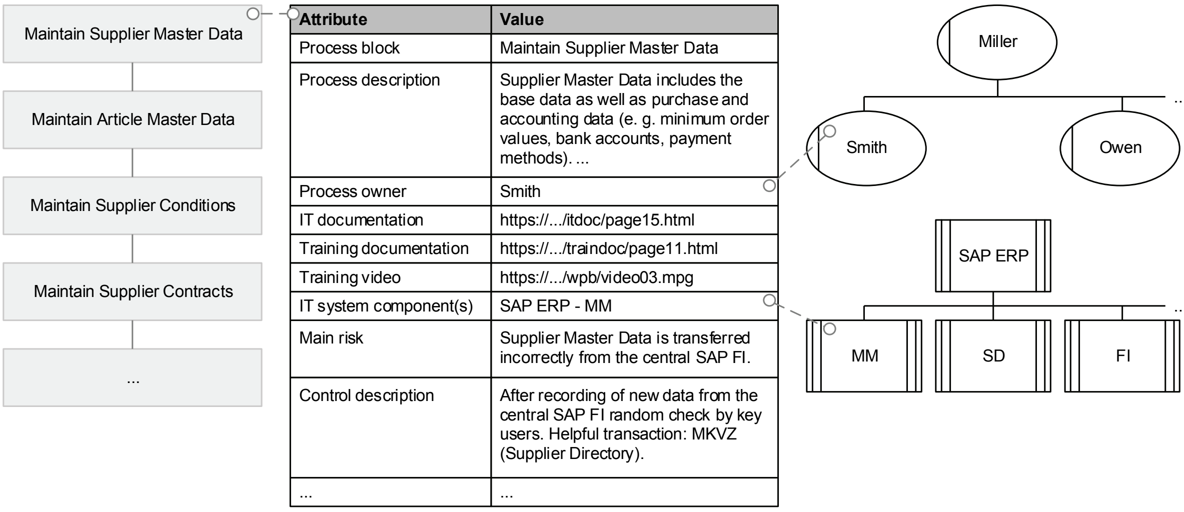
In order to communicate changes it is common sense that the content is presented in a clear and structured way. One main issue is the granularity of the business process models. On the one hand detailed descriptions of the process flow are necessary in order to understand the processes. On the other hand too detailed descriptions and therefore too complex processes are hardly understandable. The usage of different layers of abstraction is a commonly used method to cope with the complexity. With respect to communicating changes it is beneficial that all process models should follow the same levels of abstraction. Based on literature a four level structure is a good compromise regarding fast navigation, clarity, and overview. It should consist of an organizational framework (process landscape or process overview), main processes, detailed processes, and process building blocks (Becker et al., 2010; Becker et al., 2013; Harrington, 1991). Through the upper layers an executive summary for managers is given. The lower levels in the structure provide a detailed description for employees working in the departments.

These different layers are implicit within organizational charts and of IT system hierarchies, which are relevant for a full representation of all changes within the company. Organizational charts are very helpful to display all employees within the project and future working environment. Comparing the old hierarchy to the new designed one helps to communicate changes regarding the amount of positions and job descriptions. For Example park invoices will no longer be necessary because of using optical character recognition for invoices. The differences within the IT systems diagrams are of great value while estimating the need for new hard- and software.

3.3 Integrated Documentation

A full documentation of all changes is the main deliverable in change management projects. Hence, the tool has to support the user in creating the documentation. The basis for a documentation covering all relevant aspects (the scope of these has to be defined at the start of the project) are attributes within all process layers. It turned out to be beneficial, that attributes were available on all process levels and also within the organizational charts and IT systems hierarchies. Typical attributes are: generally description of process blocks, support by IT systems, or process responsibilities. The relevant position from the organizational chart or the documented IT system could be attached to process blocks directly. In the context of change management this allows a mapping of the old process responsibilities or necessary IT systems to the new ones. Also more sophisticated and ERP specific attributes like uniform resource locators (URLs) to documents stored in the SAP solution manager, direct links to the user interface, and interactive training videos add value for users.

Beside the documentation of the changes within the tool also export functionalities to office applications like Microsoft Word or Visio are


Maintain Supplier Master Data
Maintain Article Master Data
Maintain Supplier Conditions
Maintain Supplier Contracts
...
Miller
Smith
Owen
...
SAP ERP
MM
SD
FI
...

Attribute	Value
Process block	Maintain Supplier Master Data
Process description	Supplier Master Data includes the base data as well as purchase and accounting data (e. g. minimum order values, bank accounts, payment methods). ...
Process owner	Smith
IT documentation	https://.../itdoc/page15.html
Training documentation	https://.../traindoc/page11.html
Training video	https://.../wpb/video03.mpg
IT system component(s)	SAP ERP - MM
Main risk	Supplier Master Data is transferred incorrectly from the central SAP FI.
Control description	After recording of new data from the central SAP FI random check by key users. Helpful transaction: MKVZ (Supplier Directory).
...	...

Figure 3: Attributes for process blocks.

important. They facilitate the creation of tests, trainings and system documentations. In the context of the project several reports like “which processes are assigned to specific IT system components” or “which contact persons are responsible for which process steps” are especially useful. These reports are easy to create within a tool providing flexible attribute definition and reporting functionality using these attributes. The attributes are also used for the implementation of risk management. For each process step specific attributes are defined describing the risk, its’ severity and its’ possible coverage (e. g. usage of SAP transactions for Intermediate Document (IDoc) monitoring to control communication risks with suppliers). Another application area in the context of change management is creating reports with reference to specific business objects taken from the glossary, e. g. during ERP replacement the communication with suppliers is switched to electronic data interchange (EDI) protocol. Hence, looking for any old process blocks containing the activities “print” or “send” is a good starting point for managing the change. Figure 3 depicts an example where the attributes are used for connecting the operational and organizational structures.

3.4 Re-use of Knowledge

The employees from the departments provide the external consultants with all necessary knowledge in order to model the as-is and to-be processes. The re-use of single process steps with a special highlighting of those remaining identical has turned out as a driver for the acceptance of the designed process models. Therefore, it is beneficial if the BPM tool is able to use references rather than simply copying process blocks. By this means, there is no redundant data. The same information can be edited and stored in one spot and is used in potentially several process models. Figure 4 shows a referenced process block “Archive Document” used in the two processes invoicing and billing. In both cases the used archive system and all other attributes were identical and can be edited consistently.

A further facilitator for eliciting the process knowledge is the usage of reference models. During the workshops and interviews in the identification phase the external consultants proposed “best-practice” process models and organizational forms including position descriptions. It is important to propose these reference models not in the beginning but after some time of discussion in order to consolidate the knowledge. Furthermore, it is easier for all participants to adopt the existing reference models to their needs than to start from scratch. From a change management perspective the usage of the reference models yielded high quality models and allow to a certain amount benchmarking with reference companies. This is an important driver for acceptance in the communication phase.

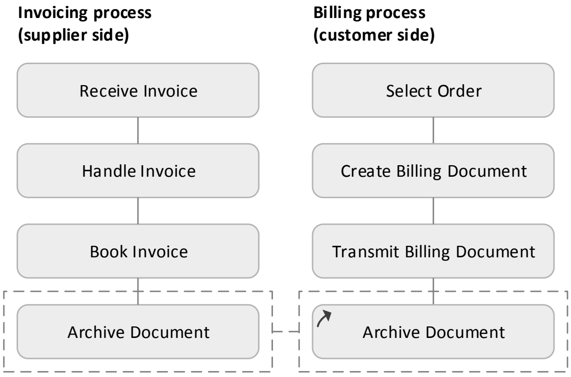


Figure 4: Process references.

Also the architecture of the tool as a web-based platform is beneficial because a quick access via web browser is possible instead of an installation hurdle. Changes could be entered during workshops directly and the user trainings could be supported by the tool since everybody was able to view the processes with additional information while using the new ERP system. Just alike, future employees will be able to use the process documentation to view and understand not only the processes they work in but also the previous and following processes.

4 CONCLUSION AND FUTURE WORK

For supporting the change management in the described project the usage of a business process modeling tool is a very promising approach. After having collected the information regarding the organizational and operational structure in the identification phase we documented and analyzed it with the BPM tool. Regarding our first research question, we conclude that a BPM tool is well suited in the context of an ERP replacement. In order to answer the second research questions, we clustered the key features in standardization of semantics, structured models, integrated documentation, and the re-usability of knowledge (Table 2).

Although the feedback within the project is positive (we are in the communication phase while writing this article), there are several potentials regarding integration, collaboration, and visualization we would like to address. We believe these do not only apply to the specific combination of used BPM tool and SAP ECC in our case. They are also relevant in comparable projects and for future BPM tool developments and research.

Integration: A strong link between the BPM tool and the new ERP system would probably increase the acceptance of the tool, e. g. a single sign on functionality.

Collaboration: Automatic version management was missing. We believe from our practical work during the project that the possibility to have a full version control of the models similar to a text document in a SharePoint environment is of great value. The version management would allow to trace the modified parts of a model, e. g. before and after a change request is implemented. Furthermore, some sort of social media integration would have been beneficial. Either to comment the different versions of the models in the review process or to interact with the process owner e. g. via chat directly. The web-based version of the tool was great for distributing the process models via a simple link. Nevertheless, in few situations during more informal meetings a mobile application would have enhanced the acceptance and would have further simplified the communication. For both solutions – the web-based and a possible mobile application – a more sophisticated user access control is needed, e. g. a guest access for a supplier. The authorization rules have to ensure that the supplier can only access the relevant and maybe new processes (e. g. transmission of shipping notifications). Similar user access control mechanisms would be relevant for the approval and monitoring of (new) process versions. E. g. processes are initially stored with the status “wait for approval”. After the new version is reviewed and released, the processes will be unlocked for all authorized employees.

Visualization: In several situations one model or different models were compared. A supporting function for a comparison of two versions of one model (e. g. as-is and the to-be) or for two IT system charts is missing but would have simplified the work for the consultant. A further aspect of the visualization is the tool visualization itself in different web browsers. Partly the tool behaved differently what definitely has to be considered when using a web-based tool in a company.

Beside the above mentioned potentials and open issues we want to continue with our evaluation until the ERP replacement is over. We are looking forward to use the BPM tool in future ERP replacements or implementations.

REFERENCES

Becker, J., Clever, N., Holler, J. & Shitkova, M., 2013. icebricks: Business Process Modeling on the Basis of Semantic Standardization. In *Proceedings of the Design Science Research in Information Systems and Technologies (DESRIST)*. Helsinki, Finland, pp. 394–399.

Becker, J., Clever, N., Holler, J., Püster, J. & Shitkova, M., 2013a. Integrating Process Modeling Methodology, Language and Tool – A Design Science Approach. In *Practice of Enterprise Modeling 2013 (PoEM 2013)*. Riga, Latvia: IFIP International Federation for Information Processing, pp. 221–235.

Becker, J., Clever, N., Holler, J., Püster, J. & Shitkova, M., 2013b. Semantically Standardized and Transparent Process Model Collections via Process Building Blocks. In *Proceedings of the 5th International Conference on Information, Process, and Knowledge Management (eKNOW)*. Nice, France, pp. 172–177.

Becker, J., Kugeler, M. & Rosemann, M., 2011. *Process Management: A Guide for the Design of Business Processes* 2nd ed., Berlin, Germany: Springer.

Becker, J., Richter, O. & El-Hawari, T., 2010. Vertriebsinformationssysteme zwischen Standardisierung und Flexibilisierung: Referenzmodelle für die Prozesse im Vertrieb. In J. Becker et al., eds. *Vertriebsinformationssysteme: Standardisierung, Individualisierung, Hybridisierung und Internetisierung*. Berlin, Heidelberg, Germany: Springer, pp. 3–18.

Becker, J., Rosemann, M. & Schütte, R., 1995. Grundsätze ordnungsmäßiger Modellierung. *Wirtschaftsinformatik*, 37(5), pp. 435–445.

Becker, J., Rosemann, M. & von Uthmann, C., 2000. Guidelines of Business Process Modeling. In W. Van Der Aalst, J. Desel & A. Oberweis, eds. *Business Process Management: Models, Techniques and Empirical Studies*. Berlin, Germany: Springer, pp. 30–49.

Breuker, D., Pfeiffer, D. & Becker, J., 2009. Reducing the variations in intra- and interorganizational business process modeling – An empirical evaluation. In *Wirtschaftsinformatik Proceedings 2009*., pp. 203–212.

Carr, N.G., 2003. IT Doesn’t Matter. *Educause Review*, 38, pp. 24–38.

Davenport, T.H., 1998. Putting the Enterprise into the Enterprise System. *Harvard Business Review*, 76(4), pp. 121–131.

Dean, D., Orwig, R., Lee, J. & Vogel, D., 1994. Modeling with a group modeling tool: group support, model quality, and validation. In *Twenty-Seventh Hawaii International Conference*. Wailea, HI, USA: IEEE, pp. 214–223.

Dean, D., Orwig, R. & Vogel, D., 2000. Facilitation methods for collaborative modeling tools. *Group Decision and Negotiation*, 9(2), pp. 109–128.

Harrington, H.J., 1991. *Business Process Improvement: The Breakthrough Strategy for Total Quality, Productivity, and Competitiveness*, New York, USA: McGraw-Hill.

Holland, C.P. & Light, B., 2002. A Critical Success Factors Model For ERP Implementation. *IEEE software*, 16(3), pp. 30–36.

Huang, S.-M., Hung, Y.-C., Chen, H.-G. & Ku, C.-Y., 2004. Transplanting the Best Practice for Implementation of an ERP System: A Structured Inductive Study of an International Company. *Computer Information Systems*, 44(4), pp. 101–110.

Mendling, J., Reijers, H. & van der Aalst, W.M.P., 2008. Seven Process Modeling Guidelines (7PMG). *Information and Software Technology*, 52(2), pp. 127–136.

Moody, D.L., 2005. Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. *Data & Knowledge Engineering*, 55(3), pp. 243–276.

Nah, F.F.-H., Lau, J.L.-S. & Kuang, J., 2001. Critical factors for successful implementation of enterprise systems. *Business Process Management Journal*, 7(3), pp. 285–296.

Ould, M.A., 1995. *Business Processes: Modelling and analysis for re-engineering and improvement*, New York, USA: Wiley Chichester.

Peslak, A.R., Subramanian, G.H. & Clayton, G.E., 2008. The Phases of ERP Software Implementation and Maintenance: A Model for Predicting Preferred ERP Use. *Journal of Computer Information Systems*, 48(2), pp. 25–33.

Rittgen, P., 2009. Collaborative modeling of business processes: a comparative case study. In *Proceedings of the 2009 ACM symposium on Applied Computing*. Honolulu, Hawaii, USA: ACM, pp. 225–230.

Scheer, A. & Nüttgens, M., 2000. *ARIS Architecture and Reference Models for Business Process Management*, Berlin Heidelberg, Germany: Springer.

Van Hee, K.M., 1994. *Information systems engineering: a formal approach*, New York, USA: Cambridge University Press.

Future Business Model for Cellular Microgrids

Intisar Ali Sajjad, Roberto Napoli and Gianfranco Chicco
Energy Department, Politecnico di Torino, Turin, Italy
malik.sajjad@polito.it

Keywords: Microgrids, Regulations, Business Model, Consumers, Entrepreneurship.

Abstract: Different studies which were carried out in the past revealed that the environment for microgrids is very complex and uncertain due to regulatory and legal barriers. Across and within the developed countries the suggestions and views of regulatory authorities and legal bindings about the infrastructure and operation of microgrids are quite different. According to the present scenario, the viability of microgrids mainly depends upon how microgrids are framed, who owns them, which are the customers served from them and how much revenue is generated from them. This paper investigates the potential barriers in current business models to deploy microgrids and proposes a business model, centric to users, with the concept of consumers owned microgrid.

1 INTRODUCTION

There are lots of uncertainties in the power system, especially for electric utility systems. These uncertainties range from capital costs and financing to fuel price fluctuations and operational costs. Due to these trends, it is very difficult to assess and formulate long term capacity planning. The ultimate impact is on consumers to compensate the additional costs through different tariff elements like fuel adjustment charges, capacity charges, etc. On the other hand, there are much more uncertainties associated with the capital costs and financing due to inflation and changing interest rates. This has affected the utilities' ability to obtain bonding for long term projects. The regulatory framework is also a key player affecting the day to day operation, together with the structure of the electricity market. In some aspects the regulation supports the utility industry to operate a stable, economic and reliable system, but in other cases it decreases the attraction of investors to invest in the system to cater for future challenges (Schweppe, Tabors and Kirtley, 1981).

The power system business has been restructured in the last decades. This was done to increase system efficiency, decrease costs and emissions, and attain reliability by using new emerging technologies in the generation, transmission and distribution sectors. Smart grid has been the unifying concept for the application of these emerging technologies. These technologies are based on the revolutionary findings in the field of power electronics, artificial intelligence, computer applications, networking abilities. These technologies have the ability to change the behavior of the electricity distribution system into an active one, where each component has the capability to talk and listen. But the goals have not been achieved yet. The progress is very slow and there is lack of investments in the field of smart grids. The major reason is the structure of the current business model in power market especially on the distribution side, more centric on utilities but less on consumers (Schweppe, Tabors and Kirtley, 1981).

Microgrids are the systems that link different distributed energy sources into a single small network and give service to its consumers with all or partial of their energy demands by increasing energy efficiency, reliability and reduce emission and energy costs (Center for Energy and Hyams, 2010). Fast growth in distributed generation, emerging ICT, power electronics technologies, efficient storage has made the dream of microgrids true and implementable. But one aspect that is not favorable, are the power market policies, regulations and legal bindings. These are very important components to attract investments from public and private sectors. One of the important reasons for the lag of such components may be the monopolization of the market, especially at the distribution level.

King (2006) assessed the different microgrid business models considering the ownership status.

According to these models, the owner is responsible for all type of services and quality indices to consumers. The owner may be a utility, a single landlord, multiple individuals or firms, a single individual, or a single firm. Center for Energy and Hyams (2010) used this concept and further categorized the business models of King (2006) using physical and virtual microgrids. Center for Energy and Hyams (2010) also gives more detailed business models for utility, non-utility, also considering virtual microgrids depending on the aggregation level.

The main theme behind the proposed business model is to split the power system network into autonomous parts that replicate the *small world* concept. This technical splitting should be followed on the same lines as the splitting of communities. This concept of small world has been used in different real world applications including power systems, transportation system, social networks, and medical science (Pagani and Aiello, 2013; Bork et al., 2004; Hidefum, 2013; Eppstein et al., 2013). Each small world should have operating autonomy and ownership by the customers with responsibility. The business model should be structured in such a way that it will attract private investments and should leave space for entrepreneurship.

The rest of the paper is organized as follows. Section 2 gives a brief overview about the relationship between innovations and entrepreneurship. Section 3 and 4 discusses about the current business models and their pitfalls. Section 5 discusses about the proposed possible vision for the future business model centric to the active consumers. This paper is concluded in Section 6.

2 INNOVATIONS AND ENTREPRENEURSHIP

Entrepreneurship is the basic idea behind the possible business model of microgrids. Shane (2003) explains entrepreneurship and gives some characteristics of an entrepreneur that introduces innovations and new gears and transforms innovations into profit and financial goods.

The role of opportunities carries central importance to establish an enterprise. Opportunities generate a situation to create means for the establishment of a profitable enterprise. Different economist and business experts explain their way to explore opportunities, with different theories.

In 1934, Schumpeter linked opportunities with innovations and said that new information is necessarily important for the existence of enterprise business. Technological, political, economic, regulatory changes provide new information to entrepreneur to recombine resources for more valuable and fruitful enterprise (Schumpeter, 1934; Urban and von Hippel, 1988). In contrast with Schumpeter, the theory presented by Kirzner states that the opportunities may need only differential access to existing information. People in a specific field use the information that they know to efficiently manage resources, but the decision making process has always shortcomings. These shortcomings lead to obtain and recombine resources for profitable business. Most of the research is done on Schumpeter's opportunities due to its diverse and innovation friendly nature (Urban and von Hippel, 1988; Schumpeter, 1934).

2.1 Consumer Centric Enterprises

The work on consumer centric enterprises started about 4 decades ago, and today it is clear that such firms and enterprises are very successful due to advances in technology, communication and computing (von Hippel, 2005).

Enos (1962) reported that the most of the innovations in the oil refining sector were developed by user companies. Also Freeman (1968) presented the results about the chemical production processes that most licensed processes were developed by user companies. More than 80% of common and important scientific instruments and semiconductor processes were developed by users. Empirical studies also proved that up to 40% of users of any services or products are engaged in developing or modifying products/services (von Hippel, 2005; Herstatt and von Hippel, 1992; Morrison, Roberts, von Hippel, 2000; Franke and von Hippel, 2003; Lüthje, 2004; Franke and Shah, 2003, Luthje, Herstatt and von Hippel, 2002).

Open source projects are one example of the above business model, where people develop products or services for themselves and share it free of cost or with very low cost to other consumers and users, e.g., open source and free software, Linux and Android applications.

2.2 Economics of Innovations

In 1957 Robert Solow presented the economic growth model based on innovation. He defined growth as the change in Gross Domestic Product

(GDP) per hour of labor per unit time (Solow, 1957). Recent prominent advances in literature give the concept of knowledge spillovers and human capital (Romer, 1986; Romer, 1990; Lucas, 1988).

3 CURRENT PRACTICES IN THE DEVELOPMENT OF MICROGRIDS

Today's scenario in the distribution system is centralized to utility and has very small or no participation of consumers.

A feedback system has been developed through smart metering, and these meters have been installed in huge quantities in some countries (Renner et al., 2011). In the past, the consumers had rarely received such kind of feedback from the operator or utility. As a result, they had a little opportunity to adapt themselves according to the need of the utility. The same case occurred for a utility to judge what kind of service (e.g., to improve continuity of supply) the consumers would really like to buy.

Research has revealed the fact that utilities may be reluctant to offer consumer centric services and price structures due to the issue of reduction in electricity sales. As far as the increasing trend of smart meters is concerned, this is only being used, in majority of the cases, to reduce cost related to billing data collection and remote load management (Kelly, Meiners and Rouse, 2007).

There are some microgrid projects like San Diego Gas and Electric's Beach Cities microgrid, Perfect power, Danish Cell Controller, AGTFTC/MCAGCC microgrid, which are implemented with participation of utilities and the private sector (GEA, 2012; GEI, 2014; Kelly, Meiners and Rouse, 2007; Becker, 2013; Russell and Sagoo, 2013). Some initiatives have also been taken in developing countries to promote the concept of smart and micro grids (Warshay, 2013).

4 PITTFALLS OF EXISTING BUSINESS MODELS

There are many participants, including utilities and consumers, in the construction of a microgrid. So, all parties are the potential candidates for the receipt of benefits obtained from microgrid services, e.g., peak load reduction. The main problem here is the definition of mechanisms for proportionate investments from participants. Also there is no adequate market on the distribution level to support microgrids (in broader sense, smart grids) to monetize benefits, and this leads to investments stranding especially from the private sector.

There are no appropriate existing regulations to compensate microgrids is case of participation in grid stability. Potential benefits of microgrid are negated in current regulatory framework, e.g., distributed generation is not allowed to manipulate peak loads (Becker, 2013). Furthermore, most of the focus in the design of microgrids in on electricity, but microgrids should be considered more widely in a multi-energy perspective.

There are many smart grid and microgrid projects implemented worldwide (as the ones summarized in Section 3), but the development is very slow. One of the main reasons for slow development is the lack of private sector interest in investments. This lack of interest is due to:

- Utility's "Big Brother" role, indicating how and when the consumers will use their appliances and processes.
- Inappropriate feed-in and net metering tariffs. Consumer's energy purchase options are very limited.
- No or very low consumer's involvement in adopting innovations and their promotion.
- Current business model doesn't allow consumers to participate in energy business and its services.
- Very little control of consumers over electricity bills.

5 FUTURE PERSPECTIVE OF BUSINESS MODEL

Potential questions of interest for the implementation of smart grid are where the money will come from and in how much time. How can the process be accelerated? Shall the consumers be involved primarily? If yes, then how? This section describes a possible future of distribution system with extended applications of computational and communication capabilities and technologies with consumer centric business model as an enterprise.

5.1 A Possible Future: Consumer Centric Enterprises

Entrepreneurship is the theory behind the concept of "cellular" microgrids, where each microgrid operator will be an independent enterprise and manage its microgrid operation and services with the help of consumer's owned firms. These services may be related to installation and maintenance services of microgrid distribution system, in buildings, houses, offices, etc. Micro financing firms will provide loans with very low interest rate to adopt innovations within microgrid premises, especially for homes and offices. Consumers will also contribute to energy needs of the microgrid through net metering and feed in tariffs. So, each unit of microgrid has an operator, providing services and micro-financing enterprises, and active consumers. All these will work together for the welfare of themselves and other participants. This complete setup is considered as a *cellular microgrid* unit. Each unit will support other adjacent units for more reliable, environmental friendly and cost effective operation of overall system. This will not only support the local distribution network, but will also be able to provide ancillary services to the grid. The pyramid shift from supplier-centric to consumer-centric is the key for competition and private sector investments. Some good analogies for this kind of change are mainframe computers to laptop computers, and conventional telephone sets to smart phones. Section 2 indicated an overall picture of enterprise business, its requirements, theories about its model and economics related to it.

Consumers will invest in emerging technologies related to communication and control to upgrade the electric grid due to their widespread adaptation. It will also lead towards commercialization of microgrids; and societies, communities, institutions and the commercial sector will be able to share their microgrid systems having local generations. These developments will lead towards new era of economic development in the field of power and energy.

5.2 The Starting Point

At present we are living in a period where each nation is fighting for the survival of its economy and social interests. Population is growing rapidly and everyone is looking for more energy resources to fulfill present and future needs. Furthermore, environmental issues need serious attention. A new economic equilibrium is needed for the existence in the war of 'survival of fittest'.

All these indicators give a hope for new business models because we are in the classical condition where a model of technological innovations can start a new stable economic wave with a creation of new paradigms in the economic market and in the society. An appropriate economic model of smart grids can be a triggering point leading towards a revolution for economic balance and social benefits, that is, the evolution of present into future.

5.3 Cellular Microgrids –Structure and Business Model

Fred C. Schweppe and his research group proposed the concept of Homeostatic Control in early 80's. Homeostatic control was founded on the following principles, presented by Schweppe, Tabors and Kirtley (1981):

- Consumer's independence;
- Two way communication and feedback between Utility and consumers.

The new model with cellular microgrids is actually based on these principles, evolving a new business model centric to consumers from the present model based on the utilities or system operators. The present smart grid structure is top down, where smart grid enables the smart cities and smart cities enable smart homes. It needs to be reversed by keeping in view the above principles. First we should made citizen smart to develop smart homes using smart innovations in technologies, then the combination of such homes will give birth to smart cities through cellular microgrids and ultimately smart grid. Such a smart home or smart city will be linked to the willingness of the city to struggle for personalized living style of its own, aimed to reach high level of sustainability and high quality of life for citizens. Figure 1 shows this paradigm for bottom up approach.

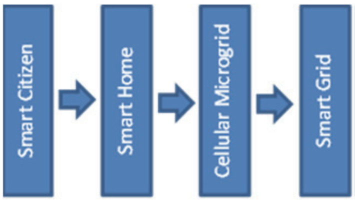


Figure 1: Bottom-up Evolution.

Smart cities will be composed of smart grids and each smart grid will have smart buildings. These micrograms will behave like cells in entire power system structure because they can communicate and interchange energy with each other and also with the grid. That's why the name proposed here is cellular microgrids.

The proposed framework is consistent with currently evolving concepts and paradigms, such as:

a) Smart Homes: The future vision of smart homes is that they will be equipped with state of the art technologies related to computation, communication and information technology. Such technologies will be able to respond to the needs of consumers (occupants) with high level of intelligence to promote their comfort, enhance convenience, security and entertainment capabilities. Through the proper management of such technologies it is possible to accommodate most of the procedures and control actions. The application of such technologies is also strongly dependent on the social attributes of occupants. For example elderly people need more comfort and young people prefer entertainment. Cost saving is also dependent on daily life events. If there is a special event e.g. birthday party then cost saving will be the secondary issue.

b) Plug and Play: the Plug and Play concept is not new and can be implemented in smart homes in future. We can get inspiration from success stories of personal computers and laptops. If we want to connect any hardware in our PC then we just need a driver to use it. Hardware of different brands can be used on single workstation due to their compatibility with a specific operating system. The same concept can be implemented here for smart homes. There should be an operating system for the power system hardware (smart appliances, protection equipment, intelligent devices, PV modules, storage devices etc.) to accommodate different manufacturer's products. Each manufacturer will provide the driver for its smart products. Smart home residents just need to insert equipment; the operating system will recognize the type of equipment automatically and will be ready to use it after the installation of its driver. Within this general scheme, the consumers can manage their needs and respond to demand response programs (e.g., spot pricing) through the intelligent system set up with the user-defined rules for demand response management.

c) Smart Generation: On the same pattern as described for smart homes, the local generation can also be made smart. It needs fewer efforts because already generation plants have some level of smartness in control and operation.

d) Structure of Self Managing Cells: Each unit of smart homes and generation can manage itself in response to internal or external signals. We call these units as cells, same concept as in human body. The structure of the biological cells is very complex but we can get some analogies for our proposed network. Each cell can communicate to other cell and can share energy on demand through the central control of all cells. We call this central control as aggregator.

e) Smart Microgrid: If we have smart homes in place, then those smart homes can be combined with each other and also with smart local generating units through a smart master control center that will take care of the needs of the individual cells. Such a system is analogous to the organism in living things. We call this organism in power system as microgrid, because microgrids have the same properties as biological organisms. Microgrids can respond to incentives, grow in size and development. Microgrids can operate in a stable way in the island mode, provided that the characteristics of the equipment connected are suitable to keep acceptable quality of supply and withstand disturbances.

f) Aggregator: The aggregator is a central controlling unit in a microgrid. It is proposed that the aggregator must be a third party other than the utility. In other words cellular microgrid should be privately owned entities so that the influence of utilities on consumers can be nullified.

5.4 Energy Pricing Mechanism (Schweppe's Optimal Spot Pricing Theorem)

The conventional social welfare objective function needs to be modified for personal optimization. Pre-determined price rates (Time of Use) do not reflect social welfare for individuals but may be useful for utilities adding pre-determined cost margins to impact on their profit.

Schweppe had presented an electricity pricing mechanism in early 80's. It is an optimal spot pricing mechanism. Detail about this theorem is presented by Caramanis, Bohn and Schweppe (1982). Here the application of spot pricing is discussed with respect to the new and evolving concept of cellular microgrids.

For the enterprise-based business model of cellular microgrids, spot pricing is proposed with price update from few hours to 5 minutes depending upon the available technology and system economics. The structure of the proposed model is shown in Figure 2. The optimal interval of updating prices is that welfare gains are equal with the additional costs due to the metering and communication resources implementation. Theory about spot pricing mechanism provides rules for the optimal decisions in the short run and also for long run actions. These actions can be taken as investments. The global social welfare function is the difference between the cost related to electricity usage and cost of generation plus investment in the overall microgrid infrastructure. Welfare depends on many factors like safety, cost, entertainment, green style. Figurative description of personal welfare function is shown in Figure 3.

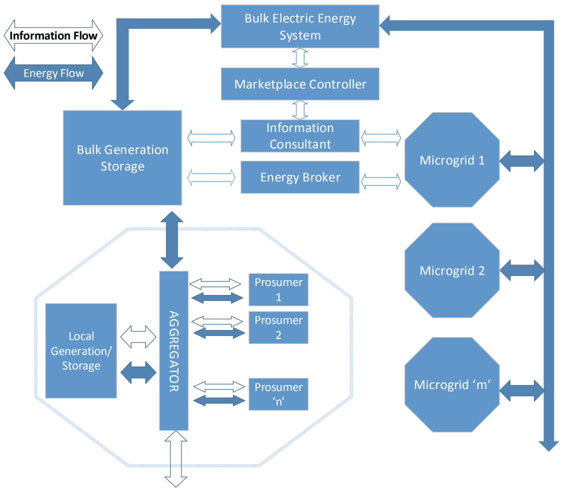


Figure 2: Information and Energy Flow in Smart Grid.

5.5 Net Metering

The concept of net metering/feed in tariffs is already implemented in some of the countries like Italy, Belgium, Canada, Greece, and Japan (REN21, 2013). It should be applied on aggregation level at the point of common coupling (PCC) to incentivize consumers and to promote local generation. The concept used in this business model is the net metering on the aggregation level at the intake of microgrid, and on a lower level for group of consumers, e.g., apartments and offices in a building, group of houses. Aggregator and building energy managers will manage internal billing mechanisms with individual cells.

5.6 Participant's Privacy

Caramanis, Bohn and Schweppe (1982) concluded that very few information of consumers are required to central control for decision making under spot pricing than time of use. Information is only related to losses, line flows and voltage overload conditions at each metering point without sharing the individual cost and profit functions. This metering point will be the PCC in case of net metering.

5.7 Business Opportunities inside a Microgrid

New business opportunities inside a microgrid, headed by the aggregator or independent of it, will be produced to support maintenance, installation services, micro financing to purchase state of the art appliances and instruments to make homes, offices, buildings and system smarter.

5.8 Social Impact

Long-term investments in the system will lead towards long-term welfare of the inhabitants residing in the area of a cellular microgrid. This welfare will be in the form of job opportunities in a local market place. This aspect has always been neglected and is very important for future investments and also to strengthen national social goals by reducing inflation.

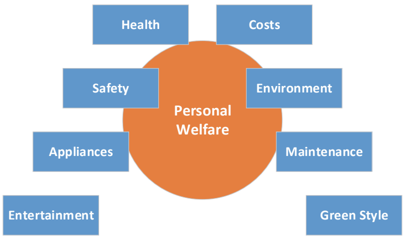


Figure 3: Personal Welfare parameters.

5.9 Regulatory Issues

Current policies and regulations are not generally favorable to user-centric business models. The Governments and the regulating bodies should be aware of the impacts of upcoming legislations that can directly or indirectly affect the business and innovations.

5.10 Is this model Evolutionary?

This model is evolutionary and can be evolved using the following initiatives:

- Net metering and feed in tariffs must be permitted and allowed for individuals and groups of consumers.
- Micro financing should be encouraged and regulatory bindings on micro financing should be relaxed for the energy business.
- Electricity bills should be totally understandable to consumers.
- Local online maintenance and installation

services should be strengthened under the aggregator supervision.

- Local skilled entities should be preferred for hiring for aggregation and other services.
- Legislation must support private sector involvement in energy business on distribution level to support, manage, own and operate a suitable size of consumers.
- Renewable energy based generation must be encouraged at the local level by incentives.
- Spot pricing mechanism must be incorporated with suitable price update time and should be decreased with innovations available.
- Distribution networks should be made more reliable by connecting consumers in meshed form, upgrading the present protection systems.
- Consumer's awareness should be enhanced through print and electronic media, and any other mean to adopt smartness.

6 CONCLUSIONS

A consumer-centric business model can be implemented using cellular microgrid structures with consumer's involvement. Small enterprises will strengthen the business structure by enhancing consumers comfort in the competitive environment. This business will also generate local job opportunities. There are regulatory issues in the implementation of such a structure, so local investment should be allowed with all kinds of feed in tariffs. Some recommendations are given in previous section to make this model more evolving. An economic study will be carried out in future work to strengthen the point of view on the cellular business model for microgrids.

REFERENCES

Becker, B., 2013 (13 March). *Microgrid Challenges: Moving Beyond Pilots*. Microgrid World Forum, Irvine. CA.

Bork, P., Jensen, L.J., von Mering, C., Ramani, A.K., Lee, I., Marcotte, E.M., 2004. Protein interaction networks from yeast to human. *Current Opinion in Structural Biology*, 14(3), 292-294.

Caramanis, M.C., Bohn, R.E., Schweppe, F.C., 1982. Optimal Spot Pricing: Practice and Theory. *IEEE Transactions on Power Apparatus and System*, PAS-101(9), 3234-3245.

Center for Energy, Marine Transportation and Public Policy at Columbia University and Hyams, M.A., 2010. *Microgrids: An Assessment Of The Value, Opportunities And Barriers To Deployment In New York State*. NYSERDA.

Enos, J.L., 1962. *Petroleum Progress and Profits: A History of Process Innovation.* MIT Press, Cambridge, MA.

Eppstein, D., Goodrich, M.T., Löffler, M., Strash, D., Trott, L., 2013. Category-based routing in social networks: Membership dimension and the small-world phenomenon. *Theoret. Computer Science*, 514, 96-104.

Franke, N., Shah, K.S., 2003. How communities support innovative activities: an exploration of assistance and sharing among end-users. *Resear. Policy*, 32, 157-178.

Franke, N., von Hippel, E., 2003. Satisfying Heterogeneous User Needs via Innovation Toolkits: The Case of Apache Security Software. *Research Policy*, 32(7), 1199-1215.

Freeman, C., 1968. Chemical Process Plant: Innovation and the World Market. *National Institute Economic Review*, 45, 29-51.

Galvin Electricity Initiative, *What are some examples of smart microgrids.* http://galvinpower.org/resources/microgrid-hub/smart-microgrids-faq/examples, 20/01/2014.

GEA Writing Team, 2012. *Global Energy Assessment: Toward a Sustainable Future*. Cambridge Univ. Press.

Herstatt, C., von Hippel, E., 1992. From Experience: Developing New Product Concepts Via the Lead User Method: A Case Study in a "low tech" Field. *Journal of Product Innovation Management*, 9, 213-221.

Hidefum, S., 2013. Structuring Hierarchical Multi-Star Small-World Networks for Real-World Applications. *Procedia Computer Science*, 18, 2410–2419.

Kelly, J. F., Meiners, M., Rouse, G., 2007. *IIT Perfect Power Prototype*. Galvin Electricity Initiative, Illinois Institute of Technology.

King, D.E., 2006. *Electric Power Micro-grids: Opportunities and Challenges for an Emerging Distributed Energy Architecture*. PhD Thesis, Carnegie Mellon University, http://wpweb2.tepper.cmu.edu/ceic, (20/01/2014).

Lucas, R.E., 1988. On the mechanics of economic development. In *Journal of Monetary Economics*, 22(1), 3-42.

Luthje, C., Herstatt, C., von Hippel, E., 2002. *The impact of use experience and knowledge on innovations: An empirical study of innovative activities among end-users*. MIT Sloan School of Management.

Luthje, C., 2004. Characteristics of innovating users in a consumer goods field: An empirical study of sport-related product consumers. *Technovation*, 24(9), 683-695.

Morrison, P.D., Roberts, J.H., von Hippel, E., 2000. Determinants of User Innovation and Innovation Sharing in a Local Market. *Management Science*, 46(12), 1513-1527.

Pagani, G.A., Aiello, M., 2013. The Power Grid as a complex network: A survey. *Physica A: Statistical*

Mechanics and its Applications, 392(11), 2688-2700.

Renewable Policy Network for the 21st Century (REN21), 2013. *RENEWABLES INTERACTIVE MAP*. http://www.map.ren21.net/, (20/01/2014).

Renner, S., Albu, M., van Elburg, H., Heinesmann, c., Lazicki, A., Penttinen, L., Puente, F., Saele, H., Feb. 2011. *European Smart Metering Landscape Report*. Österreichische Energieagentur – Austrian Energy Agency (AEA), Vienna.

Romer, P.M., 1990. Human capital and growth: theory and evidence. *Carnegie-Rochester Conference Series on Public Policy*, 32, 251-286.

Romer, P., 1986. Increasing returns and long-run growth. In *Journal of Political economy*, 94(5), 1002-1037

Russell, M. D., Sagoo, B., 2013 (September). *A Holistic Microgrid Energy Management System for Improved Energy Efficiency and Renewable Integration.* Microgrid Symposium – Santiago, Chile.

Schweppe, F.C., Tabors, R.D. and Kirtley, J.L, 1981. *Homeostatic Control: The Utility/Customer Marketplace for Electric Power*. MIT Energy Lab. Report, MIT Press, MIT-EL 81-033.

Schumpeter, J.A., 1934. *The theory of economic development: an inquiry into profits, capital, credit, interest, and the business cycle*. Harvard University Press, Cambridge, MA.

Shane, S. A., 2003. *A General Theory of Entrepreneurship: The individual-Opertunity Nexus*. Edward Elgar Publishers.

Solow, R.M., 1957. Technical Change and the Aggregate Production Function. *Review of Economics and Statistics*, 39, 312-20.

Urban, G. L., von Hippel, E., 1988. Lead User Analyses for the Development of New Industrial Products. *Management Science*, 34(5), 569-582.

von Hippel, E., 2005. *Democratizing Innovation.* MIT Press, Cambridge MA.

Warshay, B., 2013 (14th March). *Microgrids in Remote Applications: Data Centres, Developing Regions, and Military Installations: Innovative Approaches for Electrification in Developing Regions. Presentation.* Microgrid World Forum, 2013, Irvine, CA.

Comparison of Data Management Strategies for Multi-Tenant Database Cluster

Evgeny Boytsov and Valery Sokolov

Department of Computer Science,Yaroslavl State University, Yaroslavl, Russia

{boytsovea, valery-sokolov}@yandex.ru

Keywords: Databases, SaaS, Multi-tenancy, Data Management Strategies.

Abstract: This paper discusses the problem of tenant data distribution in a multi-tenant database cluster - the concept of reliable and easy to use data storage for high load cloud applications with thousands of customers, based on ordinary relational database servers. The formal statements of the problem for cases with and without data replication are given and a metric for evaluating the quality of data distribution is proposed. The proposed metric is compared with ad-hoc data management strategies using an experiment at the imitation model of the multi-tenant database cluster and the result of the experiment is provided and summarized.

1 INTRODUCTION

One of recent main trends in the software development industry is the propagation of cloud technologies and corresponding change of the main architectural paradigm in an enterprise segment of the market. This tendency leads to the increase of the software complexity, since a typical cloud application consists of tens and even hundreds distributed web-services interacting with each other. One of the most significant aspects of software design is a data-storage subsystem. This subsystem should provide high performance, fault-tolerance and reliable tenants data isolation from each other. Modern software development techniques tend to solve these tasks by designing an additional layer of application logic at the level of application servers. Such approaches are discussed in many specialized papers for application developers and other IT-specialists (Chong and G., 2006; Candan et al., 2009). This paper is devoted to an alternative concept of a multi-tenant database cluster which proposes the solution of the above problems at the level of a data storage subsystem.

One of the main challenges when implementing such a system is to choose the most efficient data management strategy which will provide the best distribution of the query flow among database servers within the cluster. In this context, the word ”best” implies a number of questions that can be answered in different ways. An optimization can be done by various criteria and we need to use some consumer characteristics to evaluate the observed quality of service. The average cluster response time, the total amount of required resources within the given service level agreements (SLA’s) or something else can be used as such characteristics. Often These characteristics are often difficult to evaluate and sometimes they conflict with each other. Besides, many of the above characteristics can be evaluated only when the distribution of clients has been already done. So far, an additional metric is required which has a direct correlation with the above consumer characteristics and can be used to find the optimal tenant distribution. This paper discusses one approach to choosing such a metric and compares its results with ad-hoc data management strategies.

2 BACKGROUND

The problem of providing a reliable and scalable data storage for cloud applications was discussed in several works. Usually, NO-SQL databases are used as cluster nodes. In particular, the problem of tenant migration in a multi-tenant environment was studied and the protocol to implement such a migration was proposed in (Elmore et al., 2011). Other researches were devoted to minimizing an owning cluster consisting of NO-SQL in-memory databases in IaaS environment (Schaffner et al., 2013; Yang et al., 2008). The algorithm of tenant distribution for minimizing expenses with respect to SLA’s was proposed in (Lang et al., 2012).

A multi-tenant database cluster (Boytsov, 2013) discussed in this paper is a concept of a data storage subsystem for cloud applications. It is an additional layer of abstraction over ordinary relational database servers with a single entry point which is used to provide the isolation of cloud application customers data, load-balancing, routing the queries among servers and fault-tolerance. The main idea is to provide an application interface which has most in common with the interfaces of the traditional RDBMS (relational database management system).

A multi-tenant cluster consists of a set of ordinary database servers and specific control and query routing servers. The query routing server is a new element in a chain of interaction between application servers and database servers. In fact, this component of the system is just a kind of a proxy server which hides the details of the cluster structure, and whose main purpose is to find an executor for a query and route the query to him as fast as possible.

The data distribution and load balancing server is the most important and complicated component of the system. Its main functions are:

- initial distribution of tenants data among servers of a cluster during the system deployment or addition of new servers or tenants;
- management of tenant data distribution based on the collected statistics including the creation of additional data copies and moving data to another server;
- diagnosis of the system for the need of adding new computing nodes and storage devices;
- managing the replication.

This component of the system has the highest value, since the performance of an application depends on the success of its work.

The flow of incoming queries of the multi-tenant database cluster can be divided into N non-intersecting and independent sub-flows for each tenant $\lambda_i, i \in 1,..,N$:

$$\Lambda = \sum_{i=1}^{N} \lambda_i \qquad (1)$$

The study of statistics on existing multi-tenant cloud applications shows that there is a significant dependency between the size of data, that the client stores in the cloud, and intensity of client query flow. The analysis of the statistics also shows that the above tendency is not comprehensive and there are clients within the cluster having the intensity of the query flow that does not match the size of the stored data. The client query flow can be divided into two sub-flows: read-only queries and data-modifying queries.

$$\lambda_i = \lambda_{i_{read}} + \lambda_{i_{write}} \qquad (2)$$

Another obvious characteristics of the query flow is an average duration μ of a query at the server. This value has a significant impact on the quality of load-balancing, since it affects the formation of the cluster total load. As we know from the queueing theory, if $\Lambda\mu > B$, where B is a bandwidth of the cluster, the cluster will fail to serve the incoming flow of requests. It is also known that intensities of incoming query flows change during the lifetime of the application, that is $\lambda_i = \lambda(t), i \in 1,..,N$.

3 THE LOAD-BALANCING PROBLEM WITH CONSTANT FLOW OF QUERIES

In the work, we discuss the load-balancing of the cluster in a case when flows of incoming queries have a constant intensity, i.e. $\lambda_i = const, i \in 1,..,N$. The solution of this problem can be considered as a solution of the general problem at the point.

3.1 Clusters without Replication

We start our discussion with clusters without data replication (that is, such clusters do not provide fault-tolerance). For simplicity, we assume that $\mu = 1$ (or, equivalently, the bandwidth of each server in the cluster is divided by μ). Let C be the multi-tenant database cluster that consists of database servers $(S_1,..,S_M)$, for each of which we know the following values:

1. $\bar{\lambda}_i, i \in 1,..,M$ - the bandwidth of the $i-th$ database server;
2. $\bar{v}_i, i \in 1,..,M$ - the capacity of the $i-th$ database server.

There are also N clients, comprising the set T, for each of which we also know two values:

1. $\lambda_j, j \in 1,..,N$ - the intensity of the j-th client query flow;
2. $v_j, j \in 1,..,N$ - the data size of the j-th client.

We call the $M \times N$ matrix D a distribution matrix (of clients at the cluster), if D satisfies the following constraints and conditions:

1. $d_{i,j} = 1$, when data of the j-th client are placed at the i-th server, and $x_{i,j} = 0$ otherwise;
2. $\forall j \in 1,..,N \; \exists! i \in 1,..,M : d_{i,j} = 1$ - the data of each client are placed at a single server;
3. $\forall i \in 1,..,M \; \sum_{j=1}^{N} d_{i,j} v_j \leq \bar{v}_i$ - the total data size at each server is less than or equal to the server capacity;

4. $\forall i \in 1,..,M\ \sum_{j=1}^{N} d_{i,j}\lambda_j \leq \bar{\lambda}_i$ - total query flow intensity at each server is less than or equal to the server bandwidth.

We call the matrix $\tilde{D}$ the optimal matrix of distribution of clients set T at the cluster C, if for a function $f(C,T,D)$ the following condition is met:

$$f(C,T,\tilde{D}) = min\{f(C,T,D) : D - distribution\, matrix\} \quad (3)$$

The function f in this definition is the measure of load-balancing efficiency among the servers of the cluster. The problem of effective cluster load-balancing in this formulation reduces to finding the optimal distribution matrix for a given cluster C, a set of clients T and a measure of efficiency f.

3.2 Clusters with Replication

The usage of a master-slave replication allows to provide fault-tolerance and gives a chance to achieve a better query flow distribution. When discussing clusters with replication, we deal with multiple data instances of the same tenant. In this case, we have to take into account division of the tenants query flow into read-only and data-modifying parts. Only the server which hosts tenants master data-instance can serve data-modifying queries.

To precise this situation, we need to add several new features into our model. First of all, we need to introduce the notion of a replication matrix. We call a $M \times N$ matrix R a matrix of replication (of tenants data instances at the cluster C) for the given matrix of distribution D, if the following conditions are met:

1. $R_{i,j} = 1$, if a replica of data of the j-th tenant is stored at the i-th server, and $R_{i,j} = 0$, otherwise
2. $\forall i \in 1,..,M\, and\, j \in 1,..,N : D_{i,j} = 1 \implies R_{i,j} = 0$ - if i-th server has a master copy of the tenant data, it can not host a tenant data replica.

Obviously, clusters with replication have the same service level requirements as its counterparts without replication. The disk capacity restriction is transformed into:

$$\forall i \in 1,..,M : \sum_{j=1}^{N} d_{i,j}v_j + \sum_{j=1}^{N} r_{i,j}v_j \leq \bar{v}_i \quad (4)$$

It is much difficult to formulate the second restriction on incoming flow intensities, since we don't know exactly the policy of query flow distribution among tenant data instances. All we can say is that all data-modifying queries are served at the master server. Read-only queries can be served either by the master server, or by slave servers, and the cluster control system is free to choose any conformant strategy. It can forward all read-only queries to the master server, using replicas just to provide fault-tolerance, it can route all such queries to replicas, somehow dividing the flow among them, or it can use an intermediate approach. These considerations lead us to the need to define an additional function:

$$shr : (C,T,D,R) \to S, \quad (5)$$

where S is a $M \times N$ matrix and $S_{i,j} \in [0,1]$. This function takes the set of servers C, the set of clients T, and the distribution of tenants data instances among servers within the cluster, which is described by matrices D and R and maps it to the matrix of the read-load share S. The read-load share matrix S has the following requirements:

1. $\forall j \in 1,..,N\ \sum_{i=1}^{M} s_{i,j} = 1$ - the read-only flow is completely distributed among tenant data instances
2. $\forall i \in 1,..,M,\ j \in 1,..,N : D_{i,j} = 0 \wedge R_{i,j} = 0 \implies S_{i,j} = 0$ - if the i-th server doesn't host data instance of the j-th tenant its load-share is equal to 0.

Having the matrix S introduced, we can formulate the flow-intensity constraint as the following:

$$\sum_{j=1}^{N} (d_{i,j}\lambda_{j_{write}} + d_{i,j}\lambda_{j_{read}}s_{i,j} + r_{i,j}\lambda_{j_{read}}s_{i,j}) \leq \bar{\lambda}_i, \quad (6)$$
$$\forall i \in 1,..,M$$

If we introduce the shorthand $load(i,j)$ as $load(i,j) = d_{i,j}\lambda_{j_{write}} + d_{i,j}\lambda_{j_{read}}s_{i,j} + r_{i,j}\lambda_{j_{read}}s_{i,j}$ then we can rewrite (6) as

$$\sum_{j=1}^{N} load(i,j) \leq \bar{\lambda}_i, \forall i \in 1,..,M \quad (7)$$

We call the combination of a distribution matrix D and a replication matrix R sustainable to the fault of k servers, if $\forall i_1,..,i_k, i_l \in 1,..,M$ the fault of servers $i_1,..,i_k$ and redistribution of the query flow among servers left will produce tenant distribution $(\hat{D},\hat{R})$, where $\hat{D}$ still conforms to the definition of the distribution matrix, and the combination $(C,T,\hat{D},\hat{R})$ still conforms to (7). In this paper, we omit the discussion on the term "redistribution of the query flow", since in general case it implies the definition of another function, which is responsible for election of a new master data instance, when the existing master data instance is placed at a failed server.

So we can finally formulate the load-balancing problem for clusters with replication and the requirement of k-faults sustainability as finding a combination of matrices $(\tilde{D},\tilde{R})$, which, together with the given structure of the cluster C, the set of tenants T and the read-load share function shr satisfies the following conditions:

1. $(\tilde{D},\tilde{R})$ corresponds to k-server faults sustainable distribution of tenants data instances
2. $f(C,T,shr,\tilde{D},\tilde{R}) = min\{f(C,T,shr,D,R)\}$ for some metric f

This problem reduces to the problem of cluster load-balancing without replication when $R = \Theta$. In this case, the function *shr* can be removed from the problem, since there is no alternative for $S = D$, which gives $load(i,j) = d_{i,j}\lambda_j$ as in (3).

4 SELECTION OF THE EFFICIENCY MEASURE

What is the best way to measure the efficiency of load-balancing among servers? Uniformity of the load is a good criteria here; therefore, the target function, which will measure this characteristics should be searched. The desired situation can be formulated in the following way: the share of a total query flow at each server should be as close as possible to the share of this server in the total computational power of the entire cluster. So, the function f can be written as follows:

$$f = \sum_{i=1}^{M} \left(\frac{\sum_{j=1}^{N} load(i,j)}{\sum_{j=1}^{N} \lambda_j} - \frac{\bar{\lambda}_i}{\sum_{i=1}^{M} \bar{\lambda}_i} \right)^2 \qquad (8)$$

With the measure of efficiency (8), the load-balancing problem becomes a special case of the generalized quadratic assignment problem (GQAP), which in turn is a generalization of the quadratic assignment problem (QAP), initially stated in 1957 by Koopmans and Beckmann(Beckman and Koopmans, 1957) to model the problem of allocating a set of n facilities to a set of n locations while minimizing the quadratic objective arising from the distance between the locations in combination with the flow between the facilities. The GQAP is a generalized problem of the QAP in which there is no restriction that one location can accommodate only a single equipment. Lee and Ma(Lee and Ma, 2004) proposed the first formulation of the GQAP. Their study involves a facility location problem in manufacturing where facilities must be located among fixed locations, with a space constraint at each possible location. The objective is to minimize the total installation and interaction transportation cost.

The QAP is well known to be NP-hard(Sahni and Gonzalez, 1976) and, in practice, problems of moderate sizes are still considered very hard. For surveys on QAP, see the articles Burkard(Burkard, 1990), and Rendl, Pardalos, Wolkowicz (Rendl et al., 1994). An annotated bibliography is given by Burkard and Cela(Burkard and Cela, 1997). The QAP is a classic problem that still defies all approaches for its solution and where problems of dimension $n = 16$ can be considered large scale. Since GQAP is a generalization of QAP, it is also NP-hard and even more difficult to solve.

The discussed multi-tenant database cluster load-balancing problem deals with tens and hundreds of database servers and tens and hundreds of thousands of tenants. Due to NP-hardness of the GQAP, it is obvious that such a problem can not be solved exactly or approximately with high degree of exactness by existing algorithm. So, we can conclude that to solve the above load-balancing problem, we need to suggest some heuristics that can provide acceptable performance and measure its efficiency and positive effect in comparison with other load-balancing strategies.

5 MODELLING OF DATA-MANAGEMENT STRATEGIES

The above measure of efficiency of cluster load-balancing is a heuristics which can be used to search for an efficient tenant distribution. But does it correlate with consumer characteristics of the cluster and lead to the better results than ad-hoc solutions, that can be written by any programmer? To answer these questions and to test the target function (8), several experiments were conducted at the simulation model of the cluster. The structure of the cluster with M database servers of different bandwidth (M is a parameter of the experiment) was generated using the modelling environment. At the initial moment, the cluster had no clients. Each experiment within the series consisted of 30 iterations with a selected combination of simulation parameters.

5.1 The Description of the Experiment

The experiment was conducted for clusters with and without replication. The model of the query flow was configured in a way which provided progressive registration of new clients at the cluster and therefore the corresponding increase of query flow intensity. Since the computational power of the cluster is limited and the total intensity of incoming query flow constantly increases, it is obvious that the cluster will stop serving queries at some point of time. It is also obvious that if one load-balancing strategy allows to place more clients than another one within similar external conditions with the similar requirements to cluster fault-tolerance, this load-balancing strategy is more effective and should be preferred in real systems.

5.2 Clusters without Replication

In this series of experiments the ratio between read-only and data-modifying queries is not important, since data replication is not used. Three load-balancing algorithms were used during the experiment.

The first algorithm tries to balance the load of the cluster by balancing the amount of clients at each server according to its bandwidth ratio. When deciding on where to host a new client, this algorithm calculates the ratio of the number of clients that are hosted on the server to the bandwidth of the server for all servers in a cluster and selects the one with the minimal ration (if there are several such servers, it randomly selects one of them). The algorithm takes into account only those servers that have enough free space to host a new client. This algorithm will be referred to as Algorithm wr1.

The second algorithm tries to balance the load of the cluster by balancing the size of data that are stored at each server according to its bandwidth ratio. When deciding on where to host a new client, this algorithm calculates the ratio of the total data size of clients that are hosted on the server to the bandwidth of the server for all servers in a cluster and selects the one with the minimal ration (if there are several such servers, it randomly selects one of them). Like the previous algorithm, this algorithm also takes into account only those servers that have enough free space to host a new client. This algorithm will be referred to as Algorithm wr2.

The third algorithm is based on the minimization of the target function (8). For the sake of simplicity, this algorithm was connected to the query generator information subsystem of the model to get exact values of incoming query flow intensities for each client. In reality, such an approach can not be implemented and values of query flow intensities should be obtained by some statistical procedures, but this approach is applicable for experimental purposes and testing the theoretical model. The main principle of the algorithm is simple: it alternately tries to host a new client at each server and computes the resulting value of the target function (8). Finally, the client is hosted at the server which gave the minimal value. This algorithm will be referred to as Algorithm wr3.

All three algorithms were tested in the same environment, that is, with the same mean of query cost and tenants activity coefficients distribution. The experiment results are given in Table 1. The first two columns show the parameters of the model and the algorithm that were used in the particular experiment. The third column shows the average amount of clients which was hosted at the cluster when the model met the experiment stop condition (one of the servers had the queue with more than 100 pending requests). The algorithm wr3 has shown better results than others for all three models.

Table 1: The results of the first experiment series for clusters without replication.

Algorithm	N. of servers	Avg. N. of tenants
wr1	7	385
wr2	7	278
wr3	7	387
wr1	9	520
wr2	9	373
wr3	9	523
wr1	15	834
wr2	15	578
wr3	15	844

5.3 Clusters with Replication

The same experiment setup was used for the case with the replication. Since the previous series of experiments showed the same results for clusters of different sizes, in this series the size of the cluster was constant and equal to 16. The ratio of query types was the main parameter of the experiment instead of the cluster size. Three load-balancing algorithms were used during the experiment. Each algorithm was configured to create two replicas of every data instance.

The first algorithm tries to balance the load of the cluster by balancing the amount of clients at each server according to the servers bandwidth ratio. This algorithm is a generalization of the Algorithm wr1 from the first experiment series. When deciding on where to host a new client and its replicas, this algorithm calculates the ratio of the number of clients that are hosted at the server to the bandwidth of the server for all servers in a cluster, and selects the one with minimal ration (if there are several such servers, it randomly selects one of them). The same procedure is applied for replicas (two in this case). The algorithm takes into account only those servers that have enough free space to host a new client or its replica. This algorithm will be referred to as Algorithm r1.

The second algorithm divides the cluster into groups of n servers, where n=Number of Required Replicas + 1 (three in this experiment series). The server with the largest bandwidth within the group is selected to be the ”master”, other $n-1$ servers become ”replicas”. When deciding on where to host a new client, this algorithm calculates the ratio of the usage for each group, and selects the group with minimal ration (if there are several such groups, it randomly selects one of them). The algorithm takes into account only those groups that have enough free space to host a new client or its replica at every server within the group. This algorithm will be referred to as Algorithm r2.

The third algorithm is a generalization of the Algorithm wr3. For every incoming request, it finds the best placement of master data instance and its replicas in terms of minimization of the function (8). The kind of branch and bounds algorithm is used to find the best solution for a current tenant. This algorithm will be referred to as Algorithm r3.

The experiment results are given in Table 2. The first two columns show the parameters of the model (ratio between read-only and data-modifying queries) and the algorithm was used in the particular experiment. The third column shows the average amount of clients which was hosted at the cluster when the model met the experiment stop condition, which was the same as in the first experiment series. The algorithm r3 has shown better results than others for all three ratios of query types.

Table 2: The results of the first experiment series for clusters with replication.

Algorithm	RO/W	Avg. N. of tenants
r1	70/30	724
r1	50/50	723
r1	30/70	666
r2	70/30	682
r2	50/50	610
r2	30/70	448
r3	70/30	564
r3	50/50	530
r3	30/70	494

6 CONCLUSION

The experiment has shown that the load-balancing strategy based on the analysis of incoming query flows intensities is more effective than ad-hoc strategies. This fact leads to the conclusion that the above theoretical concepts are correct and can be applied to construct more complicated load-balancing strategies which take into account more factors and can be used in more complicated environment. Especially interesting questions to study are:

- how to determine the incoming query flow intensity of the client in a real environment;
- what algorithms can be used to find a better solution for the clients assignment problem;
- are all solutions of the clients assignment problem equally valuable when intensities of incoming query flows are not constant;
- what strategy should be used to relocate client data when the load balancing subsystem decides to do so.

All these questions are crucial in implementing efficient load-balancing strategy for the cluster.

REFERENCES

Beckman, M. and Koopmans, T. (1957). Assignment problems and the location of economic activities. *Econometrica*, 25:53–76.

Boytsov, E. (2013). Designing and development of the imitation model of a multi-tenant database cluster. *Modeling and analysis of information systems*, 20.

Burkard, R. (1990). Locations with spatial interactions: The quadratic assignment problem. *Discrete location theory*, pages 387–437.

Burkard, R. and Cela, E. (1997). Quadratic and three-dimensional assignment problems. pages 373–392.

Candan, K., Li, W., Phan, T., and Zhou, M. (2009). Frontiers in information and software as services. In *Proceedings of ICDE*, pages 1761–1768.

Chong, F. and G., C. (2006). Architecture strategies for catching the long tail.

Elmore, A., Das, S., Agrawal, D., and El Abbadi, A. (2011). Zephyr: Live migration in shared nothing databases for elastic cloud platforms. In *SIGMOD Conference*.

Lang, W., Shankar, S., Patel, J., and Kalhan, A. (2012). Towards multi-tenant performance slos. In *ICDE*.

Lee, C.-G. and Ma, Z. (2004). The generalized quadratic assignment problem. Technical report, University of Toronto, Department of Mechanical and Industrial Engineering, Toronto, Canada.

Rendl, F., Pardalos, P., and Wolkowicz, H. (1994). The quadratic assignment problem: A survey and recent developments. In *Proceedings of the DIMACS Workshop on Quadratic Assignment Problems*, volume 16, pages 1–42. American Mathematical Society.

Sahni, S. and Gonzalez, T. (1976). P-complete approximation problems. *Journal of ACM*, 23(3):555–565.

Schaffner, J., Januschowski, T., Kercher, M., Kraska, T., Plattner, H., Franklin, M., and Jacobs, D. (2013). Rtp: Robust tenant placement for elastic in-memory database clusters. In *SIGMOD Conference*.

Yang, F., Shanmugasundaram, J., and Yerneni, R. (2008). A scalable data platform for a large number of small applications. Technical report, Yahoo! Research.

Service Composition Based on Semantic Vocabulary

YuHui Ning, ShuXia Yu, YuYue Du and Wei Liu

College of Information Science and Engineering, Shandong University of Science and Technology,
579 Qianwangang Road, Huangdao, China
{ningyuhui, syuqd, yydu001,yanchunchun9896}@163.com

Keywords: Web Service, Service Composition, Semantic Vocabulary, Concept Replacement, Service Clustering.

Abstract: With the development of Web services, semantics is introduced into Web services, and the information recognition ability can be enhanced with the semantic description of Web services. Thus, the technology of service composition and binding can be improved. At present, the main method of service composition, discovery, binding and replacement is similarity computing based on an ontology tree. However, constructing the ontology tree and deploying the concepts in it are difficult, and the operability of service composition with similarity computing cannot meet user requirements. To solve the above problem, a new method of service composition is introduced based on semantic vocabulary. The constructing methods of the semantic vocabulary, concept replacement, service clustering, and service composition are given. Finally, the validity and correctness of proposed methods are illustrated by simulations and the comparative analysis of simulation results.

1 INTRODUCTION

With the development of information technology, Web services based on XML (Extensible Markup Language) have developed rapidly (Sun and Jiang, 2008). Web services can be used distributed on a cross-platform system. At present, the service providers keep releasing many services into Internet. Although there are many choices for users, the efficiency and precision of service discovery are decreased. In order to solve this problem, the concepts of service groups (Liu *et al.* 2007), service pools (Sheng *et al.* 2009), service clusters (Deng and Du, 2013, Gao *et al.* 2006) and service communities (Quan *et al.* 2009, Liu *et al.* 2009) have been introduced. Their main idea is that the services with the similar functions should be clustered before binding a particular service to a composite service (Hu and Du, 2013). Clustering services can minimize the cardinal number for service discovery. Thus, the service discovery can be sped up.

WSDL (Web services Definition Language) has some limitations (Deng and Du, 2013). The data of services described with WSDL cannot be recognized completely by the computers. Thus, the ability of service clustering is limited.

With the research of Web services, many scholars introduce the semantics into Web services (Xiong *et al.* 2010). The information recognition ability can be enhanced with the semantic description of Web services. Thus, the service composition and binding would be improved. The current main method for service composition, discovery, binding and replacement is similarity computing based on an ontology tree (Wu *et al.* 2005). However, constructing an ontology tree and deploying the concepts in it remain a difficult task. The operability of service composition based on similarity computing (HAN *et al.* 2009, and Xie, 2011) is insufficient.

To solve the above problems, a method for composing Web services based on semantic vocabulary is introduced in this paper. The constructing methods of semantic vocabulary, concept replacement, service clustering, and service composition are given. At last, the validity and advantage of the proposed methods are illustrated by simulations.

The rest of this article is organized as follows. The overall design is introduced in Section 2. A semantic vocabulary is constructed in Section 3. The method of service clustering is given in Section 4. The service composition is introduced in Section 5. Simulations and comparative analysis are given in Section 6. Concluding remarks are made in Section 7.

2 OVERALL DESIGN

The overall design of service composition is given in this section. The advantage of the methods proposed in this work is introduced.

The procedure to realize service composition is shown as follows. Firstly, the semantic vocabulary is constructed via expert judgment (Wu, 2007 and Wang, 2008). The service concept class set and instance set are defined in semantic vocabulary. For example, a service concept class can include {Book Ticket, Charge, Grade …}, The instance of the concept "Book Ticket" can include {Boat Ticket, Air Ticket …}. Second, the descriptive concepts in Web services are replaced with the class concept based on the semantic vocabulary. The services are clustered based on the types of services. Finally, service composition is oriented to user demands.

The advantages of the methods proposed in this paper are described below. Firstly, from the procedure to the realization of service composition, the semantic vocabulary can be obtained from the expert judgment. The service concept class and concept instance can be obtained from semantic vocabulary. The descriptive concepts of services can be unified before service composition. The information recognition ability can be enhanced with the semantic description of services. Thus, the service composition and binding can be improved. Second, the types of services can be obtained from the expert judgment before service clustering. The overall design is shown in Figure 1.

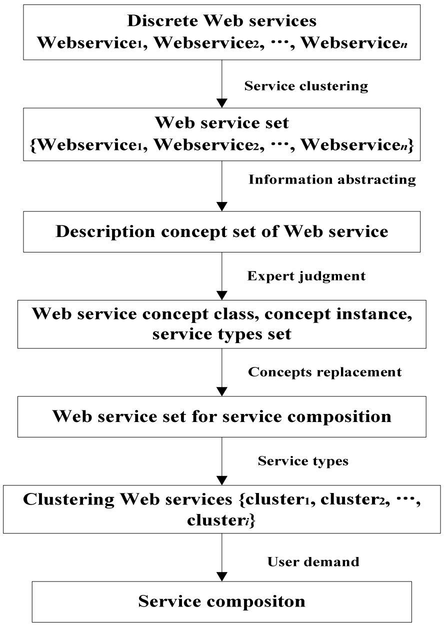

Discrete Web services
Webservice1, Webservice2, ···, Webservicen
Service clustering
Web service set
{Webservice1, Webservice2, ···, Webservicen}
Information abstracting
Description concept set of Web service
Expert judgment
Web service concept class, concept instance, service types set
Concepts replacement
Web service set for service composition
Service types
Clustering Web services {cluster1, cluster2, ···, clusteri}
User demand
Service compositon

Figure 1: The overall design of service composition.

3 CONSTRUCTION OF SEMANTIC VOCABULARY

The service and service semantic vocabulary are defined in this section. The constructing algorithm for semantic vocabulary is given next.

Definition 1. *Service*=(*No*, *Cons*) is a Web service, where

(1) *No* denotes the unique label of the service; and

(2) *Cons* denotes the descriptive concepts of the service.

Definition 2. *Ctable*=(*Class*, *Instance*, *Relation*, *Category*) is the semantic vocabulary, where

(1) *Class* denotes the service concept class;

(2) *Instance* denotes the service concept instance;

(3) *Relation* denotes the relations between concept classes and instances; and

(4) *Category* denotes the types of services.

Algorithm 1: Construction of semantic vocabulary.

Input: Service set *Tp*={$Wservice_1$, $Wservice_2$, ..., $Wservice_n$}.

Output: Semantic vocabulary *Ctable*=(*Class*, *Instance*, *Relation*, *Category*).

Step 1: Create a new semantic vocabulary *Ctable*, and *Ctable.Class*=*Ctable.Instance*=*Ctable.Relation* =$\varnothing$. Create a concept variable K=$\varnothing$. Create a one-dimension array $A[e]$, and e=+$\propto$.

Step 2: Traverse a service set *Tp*={$Wservice_1$, $Wservice_2$, ..., $Wservice_n$}.

2.1: Suppose that the current item of *Tp* is $Wservice_t$, traverse $Wservice_t.Cons$.

2.1.1: Suppose that the current item of *Cons* is Con_j, put Con_j to K.

Step 3: Traverse K.

3.1: Suppose that the current item of K is K_o, a result can be obtained via the expert judgment. If K_o belongs to the concept class, copy K_o to *Ctable.Class*.

Step 4: Copy K to *Ctable.Instance*.

Step 5: Traverse *Instance*.

5.1: Suppose the current item is $Instance_p$, traverse *Ctable.Class*.

5.1.1: Suppose that the current item is $Class_q$, a result can be obtained via the expert judgment. If $Class_q$ is the concept class of $Instance_p$, put the relation=<$Class_q$, $Instance_p$> to *Ctable.Relation*.

Step 6: Traverse *Ctable.Class*.

6.1: Suppose that the current item is $Class_q$, and there are already w types of services. A result can be obtained via the expert judgment. If $Class_q$

```
  belongs to the $r$th type of a service, and $r<w$, then,
  put $Class_q$ to $A[r]$. Else, put $Class_q$ to $A[w+1]$.
Step 7: If the number of types of services is $z$, then put $A[1]$, $A[2]$, …, $A[z]$ to $Ctable.Category$.
Step 8: Output semantic vocabulary $Ctable$.
```

Algorithm 1 presents a method for constructing semantic vocabulary. Its workflow is shown in Figure 2.

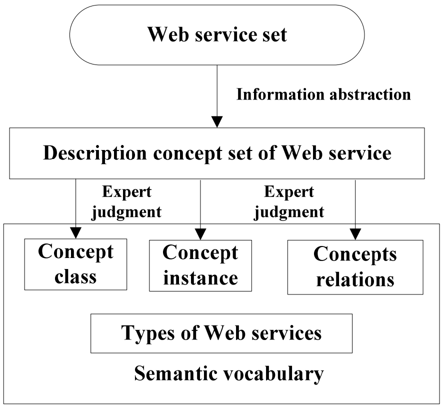


Figure 2: Workflows of constructing vocabulary.

4 SERVICE CLUSTERING

The method for service clustering is given in this section. The algorithm of service clustering is given as follows.

```
Algorithm 2: Service clustering
Input: Service set $Tp=\{Wservice_1, Wservice_2, ..., Wservice_n\}$, semantic vocabulary $Ctable=(Class, Instance, Relation, Category)$.
Output: Service cluster set $Clusters$.
Step 1: From $Ctable$, obtain the number of types of services $i$. Create a one-dimension array $A[i]=\{0\}$. Create a service cluster set $Clusters=\varnothing$.
Step 2: Traverse service set $Tp$.
  2.1: Supposing that the current item is $Wservice_j$, traverse $Wservice_j.Cons$.
    2.1.1: Supposing that the current item is $Con_k$, traverse $Ctable.Relation$.
      2.1.1.1: Suppose that the current item is $Relation_q$=<$class$, $instance$>. If $instance$ is $Con_k$, then replace $Con_k$ with $class$ in $Relation_q$.
Step 3: Traverse $Ctable.Category$.
  3.1: Supposing that the current item is $Category_r$, traverse the concepts in $Category_r$.
    3.1.1: Supposing that the current item is $Con_e$, traverse $Tp$.
      3.1.1.1: Supposing that the current item is $Wservice_j$, traverse $Wservice_j.Cons$.
        3.1.1.1.1:Suppose that the current item is $Con_w$. If $Con_w==Con_e$, then put $Wservice_j$ to $A[r]$.
Step 4: Put $A[1]$, $A[2]$, …, $A[i]$ to $clusters$, and output $clusters$.
```

From Algorithm 2, the descriptive concepts in services are unified based on semantic vocabulary. Services are clustered according to the types of services. The workflow of service clustering is shown in Figure 3.

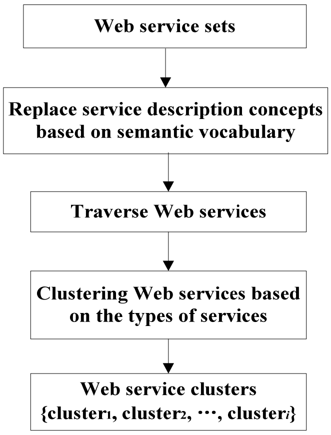

Web service sets
Replace service description concepts based on semantic vocabulary
Traverse Web services
Clustering Web services based on the types of services
Web service clusters {cluster1, cluster2, ⋯, clusteri}

Figure 3: Workflows of service clustering.

5 SERVICE COMPOSITION

The method for service composition is introduced in this section. The user requirements are defined and the workflow of service composition is described. Four types of service composition are given. The first one is the simple composition. It means that the user requirements can be satisfied by a single service. The second is sequence composition. It means that the requirements can be satisfied by some service executed consecutively. The third is parallel composition. It means that the requirements can be satisfied by some services executed in parallel. The last is mixed composition. It is the type mixed by sequence and parallel composition.

From the types of composition, the results of the service composition binding to a user are a set of services in an order. The user requirements are introduced as follows.

Definition 3. $Ur=(No, Cons, Relation)$ is a user requirement, where

(1) *No* denotes the unique label of a user requirement.

(2) *Cons* denotes the descriptive concepts of a user requirement. The structure of *Cons* is $Cons=(v, w, \ldots, k)$, where v, w, and k are descriptive concepts, and denote the atomic demand.

(3) *Relation* denotes the relations between descriptive concepts in a user requirement. The structure of *Relation* is $Relation=(x\&i\&\ldots\&p)||(q||e||\ldots||n)$, where "||" denotes logic "or", and "&" denotes logic "and".

The algorithm of service composition is shown as follows.

```
Algorithm 3: Service composition
Input: User requirement $Ur=(No, Cons, Relation)$. Semantic vocabulary $Ctable=(Class, Instance, Relation, Category)$. Service cluster set $Acluster=\{cluster_1, cluster_2, \ldots, cluster_i\}$.
Output: The result of service composition.
Step 1: Traverse $Ur.Cons$.
    1.1: Suppose that the current item is $Cons_j$, and traverse the item of it.
        1.1.1: Suppose that the current item is $Con_k$, and traverse the item of $Ctable.Relation$.
            1.1.1.1: Suppose that the current item is $Relation_q$=<$class$, $instance$>. If the $instance$ is $Con_k$, replace $Con_k$ with $class$ in $Relation_q$.
Step 2: Traverse the atomic demands in $Ur.Cons$. Suppose that the current item is $Cons_i$. Traverse $Ctable.Category$. Create one-dimension array $a[k]=\{0\}$.
    2.1: Suppose that the current item is $Category_r$. Traverse $Category_r$.
        2.1.1: Suppose that the current item is $Con_e$. Traverse $Cons_i$.
            2.1.1.1: Suppose that the current item is $Con_j$. If $Con_j==Con_e$, $a[r]$++.
    2.2: Suppose that the maximum value of $a[k]$ is a[$t$]. Traverse $cluster_t$, and set $a[k]=\{0\}$.
        2.2.1: Suppose that the current item is $Wservice_t$. Traverse $Wservice_t.Cons$.
            2.2.1.1: Suppose that the current item is $Con_w$. Traverse $Cons_i$.
                2.2.1.1.1: Suppose that the current item is $Con_j$. If $Con_j==Con_w$, $a[t]$++.
    2.3: If the maximum value of $a[k]$ is a[$m$], then replace $Cons_i$ in $Ur.Relation$ with $Wservice_m$.
Step 3: Output $Ur.Relation$.
```

The workflow of service composition is shown in Figure 4.

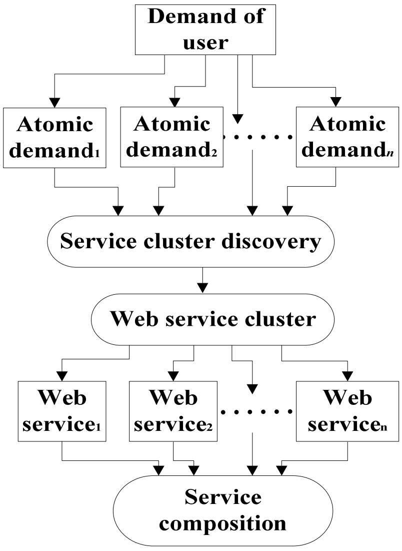


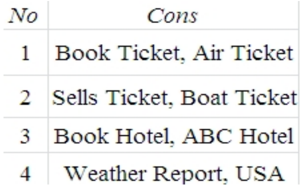
Figure 4: Workflow of service composition.

6 EXPERIMENTS

To show the advantages of the proposed methods, the experiment of service composition is given based on semantic vocabulary in this section.

(1) Construction of services

From Definition 1, 1000 services are defined in Sheet 1 of Microsoft Excel 2007. A part of services in Sheet 1 is shown in Figure 5.



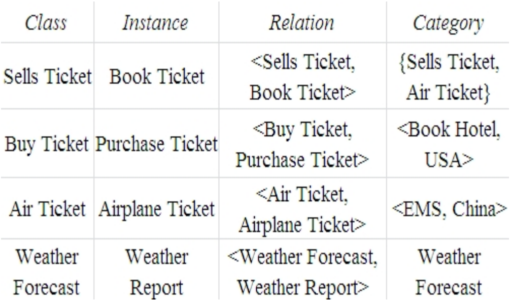
No	Cons
1	Book Ticket, Air Ticket
2	Sells Ticket, Boat Ticket
3	Book Hotel, ABC Hotel
4	Weather Report, USA

Figure 5: A part of services in Sheet 1.

From Figure 5, each line of Sheet 1 denotes a service. The unique label of a service is defined in the first column of Sheet 1, and the descriptive concepts of the service are defined in the second column. For example, from Figure 5, $Service_1=(No, Cons)$, where $Service_1.No=1$, and $Service_1.Cons=\{Book\ Ticket, Air\ Ticket\}$.

(2) Construction of semantic vocabulary

From Definition 2 and Algorithm 1, the semantic vocabulary can be obtained, and created in Sheet 2 of Microsoft Excel 2007. A part of semantic vocabulary in Sheet 2 is shown in Figure 6.



Class	Instance	Relation	Category
Sells Ticket	Book Ticket	<Sells Ticket, Book Ticket>	{Sells Ticket, Air Ticket}
Buy Ticket	Purchase Ticket	<Buy Ticket, Purchase Ticket>	<Book Hotel, USA>
Air Ticket	Airplane Ticket	<Air Ticket, Airplane Ticket>	<EMS, China>
Weather Forecast	Weather Report	<Weather Forecast, Weather Report>	Weather Forecast

Figure 6: A part of semantic vocabulary in Sheet 2.

From Figure 6, the service concept classes are given in the first column of Sheet 2, and the service concept instances are shown in the second column. The relations between concept classes and instances are given in the third column, and the types of services are described in the fourth column. For example, from Figure 6, the first type of service is presented by {*Sells Ticket*, *Air Ticket*}.

(3) Concept replacement

From Algorithm 2, the descriptive concepts of the services defined in Sheet 1 can be unified according to the service concept classes defined in Sheet 2. To improve the efficiency, the procedure for concept replacement can be programmed by Excel VBA (Visual Basic for Application) (Hu, 2014). It is convenient to extract and edit the data in Excel Sheets by VBA. The workflow of a VBA program for realizing service clustering is shown in Figure 7.

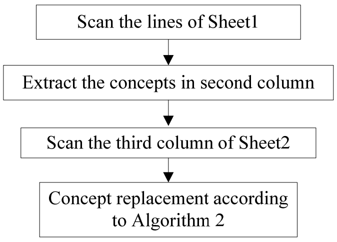
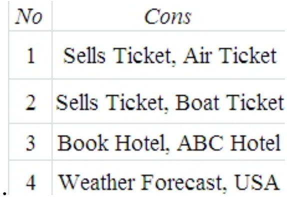

Scan the lines of Sheet1
Extract the concepts in second column
Scan the third column of Sheet2
Concept replacement according to Algorithm 2

Figure 7: The workflow of a VBA program.

Sheet 1 has been changed after concept replacement. It is shown in Figure 8.



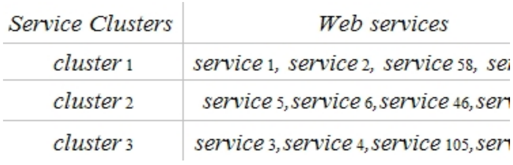
No	Cons
1	Sells Ticket, Air Ticket
2	Sells Ticket, Boat Ticket
3	Book Hotel, ABC Hotel
4	Weather Forecast, USA

Figure 8: Sheet 1 of Excel after concept replacement.

From Figures 7 and 8, the concept "*Book Ticket*" in *service*$_1$ is replaced by "*Sells Ticket*", *service*$_4$ is also changed, and so on.

(4) Service clustering

From Algorithm 2, fifty service clusters are obtained by using VBA programs, and created in Sheet 3. A part of service clusters in Sheet 3 are shown in Figure 9.



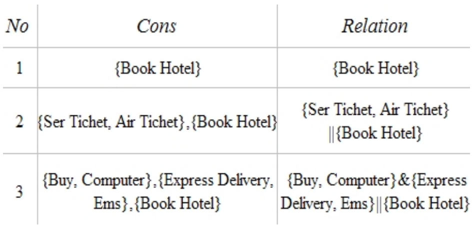
Service Clusters	Web services
cluster$_1$	*service*$_1$, *service*$_2$, *service*$_{58}$, *se*
cluster$_2$	*service*$_5$, *service*$_6$, *service*$_{46}$, *ser*
cluster$_3$	*service*$_3$, *service*$_4$, *service*$_{105}$, *ser*

Figure 9: A part of service clusters in Sheet 3.

In Figure 9, the names of service clusters are given in the first column of Sheet 3, and the services in clusters are given in the second column. For example, from Figure 9, *cluster*$_1$={*service*1, *service*$_2$, *service*$_{58}$,…}.

(5) Service composition oriented to user requirements

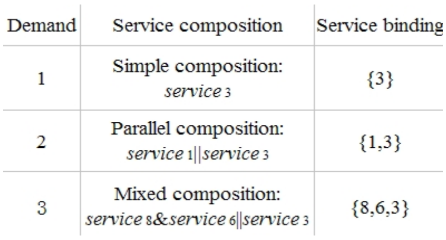
From Definition 3, fifty user demands are defined in Sheet 4. A part of user demands are shown in Figure 10.



No	Cons	Relation
1	{Book Hotel}	{Book Hotel}
2	{Ser Tichet, Air Tichet},{Book Hotel}	{Ser Tichet, Air Tichet} ||{Book Hotel}
3	{Buy, Computer},{Express Delivery, Ems},{Book Hotel}	{Buy, Computer}&{Express Delivery, Ems}||{Book Hotel}

Figure 10: A part of user demands in Sheet 4.

From Algorithm 3, a service composition can be obtained, and created in Sheet 5. A part of service composition is shown in Figure 11.

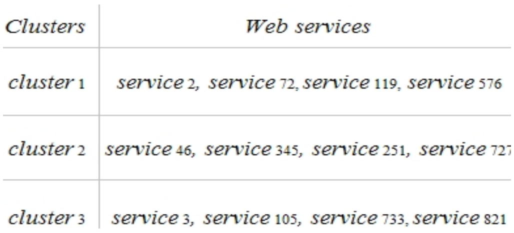


Demand	Service composition	Service binding
1	Simple composition: *service*$_3$	{3}
2	Parallel composition: *service*$_1$||*service*$_3$	{1,3}
3	Mixed composition: *service*$_8$&*service*$_6$||*service*$_3$	{8,6,3}

Figure 11: A part of service composition.

(6) Comparison

From Figure 11, the user demands can be satisfied by the service composition based on the methods proposed in this paper. The result of service clustering without semantic concept replacement is shown in Figure 12.



Clusters	Web services
$cluster_1$	$service_2$, $service_{72}$, $service_{119}$, $service_{576}$
$cluster_2$	$service_{46}$, $service_{345}$, $service_{251}$, $service_{727}$
$cluster_3$	$service_3$, $service_{105}$, $service_{733}$, $service_{821}$

Figure 12: Clusters without semantic concept replacement.

From Figures 11 and 12, obviously, the service clustering ability in Figure 12 is decreased. For example, the descriptive concepts "*Book Ticket*" in *service*$_1$ are not a concept class. The computer cannot recognize them from the service clustering. Thus, *service*$_1$ does not appear in *cluster*$_1$ of Figure 12. Moreover, the service composition ability can be decreased.

Therefore, the validity and advantages of the proposed method are illustrated.

7 CONCLUSIONS

To improve service composition, a new method of service composition is proposed based on semantic vocabulary in this paper. A semantic vocabulary is constructed and some algorithms are presented such as service concept replacement, service clustering, and service composition. Finally, the validity and advantages of the proposed methods are illustrated by simulations and comparative analysis.

Further work will be the data processing technology of the services oriented to big data. Moreover, the platform construction of service composition will be considered based on cloud computing.

ACKNOWLEDGEMENTS

This work is supported by the National Basic Research Program of China under grant 2010CB328101; the National Natural Science Foundation of China under grants 61170078 and 61173042; the Doctoral Program of Higher Education of the Specialized Research Fund of China under grant 20113718110004; Basic Research Program of Qingdao City of China under grant No. 13-1-4-116-jch; and the SDUST Research Fund of China under grant 2011KYTD102.

REFERENCES

Deng, S.Y., and Du, Y.Y., 2013. Web service composition approach based on service cluster and Qos. *Journal of Computer Applications*. 33(8), 2167-2170.

Deng, S.Y., and Du, Y.Y., 2013. Web service composition approach based on service cluster and Qos. *Journal of Computer Applications*. 33(8), 2167-2170.

Gao, Y., Na, J., Zhang, B., Yang, L., 2006. 3-layer Web services organization model for dynamic service composition. Mini-Micro Systems, 27(10), 1879-1882.

Hu, Q., Du, Y.Y., 2013. Service architecture and service discovery oriented to service clusters. *Journal of Computer Applications*. 33(8): 2163-2166.

Hu, X., 2014. Automatic transmission of delivery information based on GPRS. Journal of Neijiang Normal University, 18(9): 107-109.

HAN, S., WANG, H.Y., CUI, L.Z., 2009. A user experience oriented service discovery method with clustering technology [C]// Second Int. Symposium on Computational Intelligence and Design, China, Chang Sha: *IEEE Computer Society*, 64-67.

Liu, S.L., Liu, Y.X., Zhang, F., Tang, G.F., 2007. A dynamic Web services selection algorithm with Qos global optimal in Web services composition. *Journal of Software*, 18(3), 646-656.

Liu, X.Z., Huang, G. and Mei, H., 2009. Discovering homogeneous Web service community in the user-centric Web environment. *IEEE Transactions on Service Computing*, 2(2), 167-181.

Quan, Z.S., Boualem, B., Zakaria, M., Anne, H.H.N., 2009. Configurable composition and adaptive provisioning of Web services. *IEEE Transactions on Services Computing*, 2(1), 34-49.

Sun, P., Jiang, C.J., 2008. Using service clustering to facilitate process-oriented semantic Web service discovery, *Chinese Journal of Computers*, 31(8), 1340-1353.

Sheng, Q.Z, Benataliah, B., Maamar, Z., 2009. Configurable composition and adaptive provisioning of Web services. *IEEE Transactions on Service Computing*, 2(1), 34-49.

Wu, J., Wu, Z.H., Li, Y., Deng, S.G., 2005. Web service discovery based on ontology and similarity of words. *Chinese Journal of Computers*, 28(4), 595-602.

Wu, J.Y., 2007. Requirements and configuration management of project. *Microcomputer & its Applications*, S1(2):112-114.

Wang, X.P., Wang, F.J., Wang, W.W., 2008. The certification research of enterprise quality management system. *Management Review*, 10(8): 131-134.

Xiong, P C, Fan, Y., Zhou, M.C., 2010. A Petri net approach to analysis and composition of web services. *IEEE Transactions on Systems*, Manand Cybernetics, Part A, 40(2): 376–387

Xie, L.L., 2011. Ontology-based semantic web services cluster and discovery. *Tianjin University*, China.

Reflecting on the Ambient Intelligence Vision
A Cyber-Physical-Social Perspective

Olga Murdoch, Michael O'Grady, Rem Collier and Gregory M. P. O'Hare
CLARITY Centre for Sensor Web Technologies, University College Dublin, Belfield, Dublin 4, Ireland
olga.murdoch@ucdconnect.ie, michael.j.ogrady@ucd.ie, rem.collier@ucd.ie, gregory.ohare@ucd.ie

Keywords: Ambient Intelligence, Cyber Physical Social Systems, Middleware, Web Technologies, Ubiquitous Computing.

Abstract: By acquiring and reasoning about user and environmental context, Ambient Intelligence (AmI) systems enable intelligent and intuitive interactions between people and their physical environments. AmI traditionally seeks to build on ubiquitous sensing technologies and communications to acheive this aim. Over a decade later, people are frequently immersed in or engaged with alternative, software, environments, such as those provided as web applications or social networks. Now, to form more complete contextual representations of users, AmI solutions require the context of multiple environments in which users may be immersed. This paper proposes a middleware supported framework for cyber-physical-social AmI that relies on the combined efforts of modern web technologies and ubiquitous computing as the enabling technologies.

1 INTRODUCTION

In a modern data-saturated world, knowledge and inference-driven systems integrate intelligence into real-time services and devices allowing for the provision of diverse, self-adaptive, autonomous, personalised and intelligent computing applications (Weiser, 1991). This is the vision of ubiquitous computing research which aims to provide unobtrusive deployments of technology enabling the provision of Ambient Intelligent (AmI) solutions. AmI (Sadri, 2011) enables autonomous environments and, specifically, intuitive interaction through the provision of Intelligent User Interfaces (IUIs) enabled by in-situ intelligence and embedded decision making.

While Wireless Sensor Network (WSN) and associated middleware research focuses on enabling this goal, the World Wide Web (WWW) has become a modern day source of personal and informative, real time and archival data. Through user participation in social networks and use of personalised web services, the social web (Chi, 2008) has become an extension of people's everyday lives. Social and non-social web sites provide real time reports of topical information, breaking news and environmental information such as weather reports and regional disaster alerts. Furthermore, virtual and augmented reality research is beginning to enable real-time interactions with environments that are very different to the physical world we live in.

Acknowledging these advances in technology requires us to extend our understanding of context, environments and sensors. The enabling technologies for AmI must now accommodate modern environments and leverage advances in WWW research. Further, the implications of bridging the cyber-physical-social divide must be assessed so as not to limit user adoption of cyber-physical-social AmI solutions.

This paper summarises advances in Ubiquitous Computing and the evolution of the World Wide Web (Section 2). We then reflect on the traditional AmI vision (Section 3), offering a current perspective on what we consider *context*, *environment*s and *sensors*, and addressing the benefits and challenges of cyber-physical-social AmI. This analysis motivates our proposed middleware-supported framework for cyber-physical-social AmI, which composes existing research efforts into a unified architecture (Section 4). Conclusions and a research agenda are offered to deliver the foundations of cyber-physical-social AmI (Section 5).

2 THE WORLD WIDE WEB AND UBIQUITOUS COMPUTING

The World Wide Web and Ubiquitous Computing are diverse yet converging areas of research, both of which strive to produce personalised and intelligent solutions that will be adopted by and meet the needs of users. This section provides an overview of state of the art research in these areas followed by analysis of the supporting frameworks that contribute to our unified research agenda.

2.1 World Wide Web

Since the birth of the World Wide Web (WWW) in 1990 it has been transformed from a web of documents to a web of interconnected information, users and services (Web 2.0). The emergence and wide spread adoption of social networks in particular has embedded users into the web by providing a mechanism through which they can connect and share. By providing programmatic access to data through standardised interfaces, this social web facilitates the web of services, enabling enhanced and personalised applications which further enhancing a user's web-based experience.

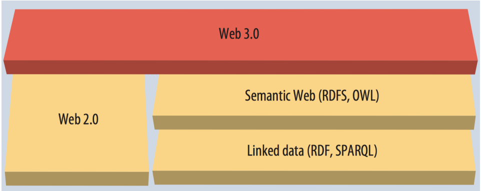


Figure 1: Semantic Web 3.0. (Hendler, 2009).

In 2009, Hendler proposed that Web 3.0 will be in part the realisation of the semantic web (Hendler, 2009) as outlined by Tim Berner's Lee in 2001 (Berners-Lee et al., 2001). It is stated that Web 3.0 can be viewed as “Semantic Web technologies integrated into, or powering, large-scale Web applications”. This is illustrated in Figure 1 which identifies the underlying semantic technologies that link Web 2.0 data allowing the formation of a Semantic Web upon which Web 3.0 applications will strive. Web 3.0 applications differ from those we know to be Web 2.0 in that they will become ‘smarter’ due to the ease of performing simple inferences with the support of semantic knowledge bases.

In 2010, Hendler and Berners-Lee (Hendler and Berners-Lee, 2010) argued that the social connected nature of current web systems is an early realisation of web connected social machines that will achieve much more through collaborative functionalities. It is argued that the future web must “be designed to allow the virtually unlimited interaction of the Web of people”. It is envisioned that people will build and share their own social machines.

While social machines have not yet been realised, Web 2.0 persists in the mainstream and Web 3.0 applications are beginning to emerge in the research literature (Hendler, 2009). Initial insights into Web 4.0 and beyond are now offered in research literature. Aghaei (Aghaei et al., 2012) and Kambil (Kambil, 2008) speculate, independently, on what is to come after Web 3.0, agreeing that Web 4.0 will be a realisation of the WWW where intelligent interactions between humans and machines take place. Kambil offers further insights into Web 5.0, proposing a sensory-emotive web, a WWW that is responsive to users emotional state or receptivity to specific information.

2.2 Ubiquitous Computing

Ubiquitous computing involves building highly responsive and self-adaptive systems which aim to increase the integration of technology into the fabric of everyday living (Weiser, 1991), for example, smart spaces and environmental monitoring. Enabling this vision requires the availability and use of heterogeneous sensing devices that enable data to be captured in real time. AmI by its very nature requires that sensors and services be unobtrusive to a user, necessitating the provision of sensors that can be embedded in the environment and existing user devices is essential.

Smart spaces aim to provide ubiquitous (hidden) services such as ambient personalised displays, mood lighting, and automated services by embedding intelligence in devices within a utilised space such as a home or office. Internet of Things (IoT) and Wireless Sensor Network (WSN) researchers aim to enable smart spaces through small to medium size deployments of smart things and sensors within an occupied space.

Environmental monitoring systems typically gather a diverse range of data relating to the conditions of an environment enabling the development of intelligent ecosystems that react to diverse phenomena such as pollution, wildlife migration, natural disaster or intrusion. Such systems require large scale WSNs that can be left in the environment unattended for long periods of time, meaning the provision of sensing devices capable of intelligence is required. The solution offered by the Sensor Web is “web accessible sensor networks and archived sensor data that can be accessed and discovered using standard protocols and Application Programming Interfaces (APIs)” (Botts et al., 2008).

Internet of Things (IoT), Web of Things (WoT) and Sensor Web are terms often given conflicting definitions. Some IoT definitions (Atzori et al., 2010) focus on internet enabled devices, while others emphasise the networking capability of such devices, not unlike the WSN's that underpin Sensor Web research. This seems a natural progression from the IoT vision that started in the RFID community where internet enabled devices are limited in computational capabilities. IoT researchers have since begun experimenting with more powerful devices or smart objects that are capable of inter-device communication amongst other capabilities such as semantic reasoning and actuation. Web of Things (WoT) research, which is seen as the next stage of the IoT, focusses on use of emerging Web Standards to facilitate reuse of internet enabled devices. The Sensor Web (Delin, 2002) paradigm envisions connectivity of entities and users in real and web-based environments through the adoption of large scale heterogeneous sensor-actuator networks.

Such networks support intelligent sensing and actuating sometimes through the embedding of agents (Tynan et al., 2005) sometimes even advocating Autonomic Wireless Sensor Networks (Marsh et al., 2004). They treat all entities as equal citizens regardless of their origin, form or nature (O'Hare et al., 2012). While IoT enabled applications described in research tend to be 'local', such as smart spaces, Sensor Web researchers strive to solve 'global' problems such as environmental monitoring and early warning systems.

2.3 Supporting Frameworks

Middleware and programming frameworks aim to ease the process of developing and deploying systems that typically compete for resources and are dependent on complex underlying technologies. While the two terms are often used interchangably within research literature, here we provide a distinction. Middleware frameworks ease and manage the deployment of underlying technologies that enable data access and management, and also facilitate application deployment and access to resources. Programming frameworks enable rapid development of user-facing prototypes and applications through the provision of API's that enable abstracted access to the data managed by the middleware. An end-to-end support framework comprises both middleware and programming technologies. Each of WWW, IoT, WSN, and SensorWeb research domains employ such supporting frameworks to allow developers of user facing solutions to remain application domain experts, without concern for the underlying infrastructure upon which their application runs.

Given the nature of the current WWW, which can be described as Web 2.0, a Social Web, and a Web of Services, Service Oriented Architectures (SOA) have become widely adopted as the supporting framework for Web application developers (Issarny et al., 2011). Service Oriented Middleware for the WWW usually focuses on service discovery and composition (Milanovic and Malek, 2004) while programming support is typically provided through mashup tools that allow developers use mutliple diverse services within a single application through an easy to use interface (Grammel and Storey, 2010). Cloud Computing infrastructures have emerged as a powerful tool for supporting rapid deployment of services and applications (Armbrust et al., 2010).

Middleware and programming frameworks for ubiquitous computing aim to ease the process of deploying, potentially large scale, WSNs, enabling developers to build data-driven solutions without the learning curve involved with network deployment and programming. There is a wealth of middleware research in the broad ubiquitous computing and WSN domains (Raychoudhury et al., 2013) (Wang et al., 2008) (Hadim and Nader, 2006), much of which provides solutions to distinct problems in specific domains, or only supports the use of homogenous sensors. A generalised middleware for inherently heterogeneous environments was called for in 2001 (Geihs, 2001); over a decade later, it has still not been realised. Due to the flexibility it offers in terms of reusability, extensibility and interoperability of services, SOA is becomming a standard approach for developing middleware solutions for the Internet of Things, WSN and Sensor Web research (Mohamed and Al-Jaroodi, 2011) (Chu and Buyya, 2007). SOA partially overcomes the heterogeinity problem bringing middleware researchers closer to realising a generic middleware solution for diverse systems.

3 REFLECTING ON THE AmI VISION

Many of the developments described in the previous sections have occurred within the last decade. In the case of AmI, the question arises as to what are the implications for paradigm itself and how or if it should reinvent itself in light of these developments. While the challenges of WWW and ubiquitous computing research differ, research and developments into the supporting frameworks for such systems are converging towards common, SOA, solutions and a single unifying framework is expected to emerge. The divide between WWW (cyber-social) and ubiquitous computing (physical-social) will be bridged, enabling a platform for cyber-physical-social AmI solutions. Bridging the cyber-physical-social divide is of benefit to the AmI paradigm. How it is accomplished must be accommodating of the users that will adopt AmI technologies and are core to the success of AmI research. Here we discuss cyber-physical-social context, environments and sensors and explore both the benefits and challenges of bridging the cyber-physical-social divide within the context of AmI research.

3.1 Cyber-Physical-Social Context, Environments and Sensors

Context comprises any information that characterizes the situation of any person, place, or object that is considered relevant to the interaction between a user and an application (Abowd et al., 1999). For AmI, context is typically aquired through deployments of ubiquitous sensing technologies monitoring users, their immediate physical environment and the things they interact with. In keeping with this definition of context, cyber-physical-social AmI requires a broader understanding of environments and sensors.

An environment can be considered to be the geographical location and immediate surroundings within which a user or entity currently resides or equally the cyber (web-based/software) environment that a user or entity is engaged with or immersed in. Modern AmI users are increasingly likely to be immersed in environments alternative to the one they are physically present in, such as social networks, games, e-commerce, and virtual/augmented reality. This must be accommodated by a sensing infrastructure enabling cyber-physical-social AmI.

While sensors are typically defined by the features of the device that enable it to sense, here we refocus on the capabilities regardless of the form of the sensor. A cyber sensor is a software sensor that monitors any programmatically accessible environment, user or entity (O'Grady et al., 2013). A sensor that is intelligent, as distinct from operating continuously in a *sense-transmit* cycle, can monitor its own state and that of connected sensors. It can also decide when it is appropriate to report a phenomenon, reducing communication cost which can have a significant impact on the operational lifespan of a battery-powered sensor, or reducing traffic to a server which is being queried by a cyber sensor. A supporting infrastructure for cyber-physical-social AmI must treat cyber sensors and physical sensing devices as equal citizens.

3.2 Cyber-Physical-Social AmI: Challenges and Benefits

A core challenge for AmI has traditionally been user adoption of AmI solutions. The importance of intuitive interaction is as important now as it was when Weiser proposed the original ubiquitous computing vision, yet remains a distant goal. In the case of AmI, if the emphasis on the physical element, that is ubiquitous computing, is augmented with a social perspective, this may inform the behaviour of AmI systems and make them more human-centric. Cyber-physical-social AmI solutions will be based on a more complete contextual foundation of its users than the knowledge base achievable through embedded sensors alone. Difficulties in assessing a user's mood or motivations will be addressed through the inclusion of a user's real time social web presence within the solution.

The traditional vision of calm computing encompasses technologies that stay hidden and make decisions without any user input. Motivated by human ability to exploit their environments, Rogers (Rogers, 2006) describes an alternative goal to Weiser's calm computing that is *engaging computing*, shifting from 'proactive computing' to supporting 'proactive people'. It is also argued that we "simply don't do 'smart' well yet" (Greenfield, 2006), but perhaps that is because we don't have access to comprehensive historical data for individual users such that a system may learn to adapt appropriately. It is proposed here that the provision of engaging technologies is a necessary step in the journey to ultimately providing calm computing. As users become more engaged with (and accepting of) technology, they will learn to understand and trust in technological advances. Simultaneously, researchers will learn what users are willing to accept from technology, both in terms of what tasks/decisions they are willing to hand over to automated solutions and what level of control users demand over their own data. This intermediate step between engaging and calm computing will facilitate the organic growth of large user annotated data sets (aggregated through user interactions with technology), providing a new historical knowledge base upon which local, personalised, intelligent solutions can become automated to suit AmI users.

While it is envisaged that middleware will bridge the cyber-physical-social divide for AmI solutions, it must be considered how that middleware will operate. While services exist that facilitate the sharing of WSN, or Internet of Things, data to the web, the publication of private sensor data to an external data broker will lead to concerns surrounding user privacy and ownership of data. The idea that all user information, including that originating from sensors embedded in private environments, is managed by third party data brokers goes against modern social expectations and will lead to resistance in terms of adopting these new technologies. Advances in information retrieval, dissemination and capture technologies far exceed those in areas of web data providence and privacy. A contributor of information to the current WWW must understand the privacy controls of hugely diverse service providers whose terms and conditions may change at any point. Relying on these external data brokers to control and secure information means users cannot be certain of who has access to their information and for what purpose. It is therefore necessary when bridging the cyber-physical-social divide, to provide a safe place where information processing and intelligent decision making can take place while adhering to user defined privacy requirements. This results in the requirement for web-based data to be integrated with WSN data within a local distributed middleware that supports fully customisable privacy and security solutions. Such an approach will facilitate fusion of widely diverse user data, integrating WWW, Sensor Web and IoT research efforts to date, as proposed in the following section.

4 PROPOSED CYBER-PHYSICAL-SOCIAL ENABLEMENT FOR AmI

In Section 2, Service Oriented Architecture (SOA) was identified as a common approach to enabling WWW, IoT and Sensor Web solutions. Despite the differences between WWW and ubiquitous computing, a mapping can be made between the challenges to be addressed by services, enabling a unifying solution to bridge the cyber-physical-social divide. An adaptation of a layered SOA architecture for cyber-physical-social enablement is described as follows (adapted from (Atzori et al., 2010)):

Objects. Sensors, Smart devices, WSNs, Targeted aspects of any software including WWW sites and services

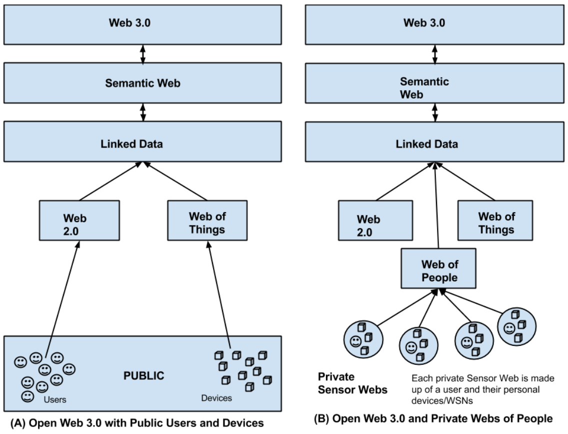
Object Abstraction. Software the unifies access to and control of objects regardless of original form, converting sensed data to a standardised format

Service Management. Software services that provides features to simplify deployment and management of a complete infrastructure, e.g., standards, object discovery, configuration and status monitoring. Trust, privacy and security, if not inbuilt to the core of the middleware architecure, are typically provided as services that can be configured as required

Service Composition. Supports the application developer in composing applications supported by the infrastructure through the composition of predefined services exposed through an API

Applications. Multiple, concurrent appplications competing for infrastructure resources in a way that is managed by the previous layers according to its configuration

A supporting framework for cyber-physical-social AmI will accommodate semantic reasoning and diverse artificial intelligence solutions (Murdoch and Nixon, 2010) enabling the development of engaging AmI solutions through which user feedback will support enhancement of automated services and more intuitive interactions.



(A) Open Web 3.0 with Public Users and Devices

(B) Open Web 3.0 and Private Webs of People

Figure 2: (A) Open Web 3.0 and Web of Things (B) Private Webs of People co-existing with Open Web 3.0.

To further understand the requirements of an enabling framework for cyber-physical-social AmI, we consider the wider perspective, how IoT, Sensor Web and WWW research agendas will unify to acheive this goal. In Section 2 we identified current and projected future trends in WWW research. In Figure 2 (A) we extend Figure 1 to illustrate how the Web of Things integrates physical sensor data in a manner that is separate to the user contributed information obtained via Web 2.0. This separation of concerns is not a true representation of the reality which involves interactions between people and things resulting in production of data. A Web of People enabled by social machines is an innovative vision that has the potential to unlock a realm of previously unimaginable personalised applications and services. However, the notion of an open social web and the current architecture of web technologies is in conflict with the necessity to provide user friendly yet highly granular customisable privacy controls.

The open WWW will only contain information that is contributed with the intention of being open and reused. We propose that the answer is not exclusively in finding better privacy solutions but in understanding how, why and when to use the open WWW. Then, we need to facilitate the evolution of personal social machines that are created from the ground up. People will be empowered to be their own data broker, privacy rules for individual pieces of data will travel with that data as it is shared/disseminated through networks of users. Data that is not deemed public will not be shared to the open web. In Figure 2 (B) we extend the Web 3.0 framework to allow for personal sensor webs, enabling a web of people. A sensor web is a collection of sensors, in this case, monitoring all entities relating to a particular user and their environment. A private sensor web comprises a user, their devices, their personal environments, and information about their connections and connected environments to which the users have been granted mutual access. It is proposed that a private sensor web will be the enabling technology for personal social machines with fine grained user control over privacy and organically grown connections between users supported by distributed mobile social networks. The separation of concerns between private user information and that shared publicly via Web 2.0 and the Web of Things will allow for bespoke privacy mechanisms that are user controlled, while the open web remains unthreatened.

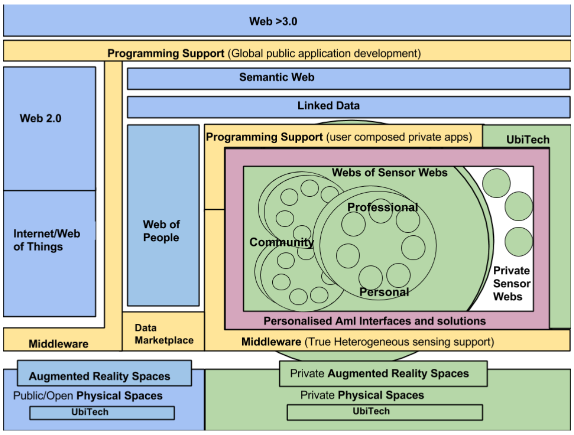

Web >3.0
Programming Support (Global public application development)
Semantic Web
Linked Data
Web 2.0
Internet/Web of Things
Web of People
Programming Support (user composed private apps)
UbiTech
Webs of Sensor Webs
Professional
Community
Personal
Private Sensor Webs
Personalised AmI Interfaces and solutions
Middleware
Data Marketplace
Middleware (True Heterogeneous sensing support)
Augmented Reality Spaces
Public/Open Physical Spaces
UbiTech
Private Augmented Reality Spaces
Private Physical Spaces
UbiTech

Figure 3: Middleware Enablement for Cyber-Physical-Social Web Greater Than 3.0.

Figure 3 further illustrates the proposed composition of ubiquitous and web technologies that will provide a foundation for future AmI solutions. Requirements for middleware and programming support are highlighted throughout as is a layer depicting user-centered, personalised, AmI solutions and interfaces. It is envisaged that private sensor webs with connecting users will form organically grown communities, or webs of sensor webs. Privately sensed information from a user's physical or web-based environments as well from publicly accessible data from Web 3.0 drive user-focussed services that adapt autonomously to user needs as they change over time. Private, community, regional and global sensor webs will underpin future smart spaces and environmental monitoring systems. It is anticipated that a data marketplace will facilitate users *selling* high level abstracted data to companies and service providers. The difference in this model is that the data resides with the user until they actively choose to share it. This will enable the provision of data providence visualisations that allow a user to understand how, where, when and why their information is being used, empowering them to make safer choices in terms of sharing private information.

Within the context of this unified cyber-physical-social web, middleware is required to bridge, seamlessly, existing and future infrastructures including WWW and Ubiquitous Computing deployments. Programming Support is required to unify access to the underlying infrastructures simplifying the development process for application domain experts.

5 CONCLUSIONS AND RESEARCH AGENDA

In this position paper, we considered AmI as a Cyber-Physical-Social paradigm. Having reflected on current state of the art in ubiquitous computing and web research, a future cyber-physical-social web was proposed as the contextual foundation that will underpin Cyber-Physical-Social AmI. The need for a generic middleware and software framework to enable this vision of AmI was articulated and Service Oriented Architecture identified as the appropriate solution for acheiving this aim. Based on this discussion, the following research priorities are identified:

- A deeper understanding of the need for cyber, in particular social, data within ambient intelligence solutions;
- A seamless and unified abstraction model of cyber (including social) and physical environments;
- Unified abstractions for middleware services supporting both WWW and ubiquitous computing initiatives
- An intuitive and feature rich middleware and programming framework for the rapid development of prototypes and testbeds.

This research agenda depends on collaborative efforts of researchers in the broad fields of WWW and Ubiquitous Computing research to deliver an infrastructure for user-centered cyber-physical-social AmI.

ACKNOWLEDGEMENTS

This work is supported by Science Foundation Ireland (SFI) under grant 07/CE/11147 and the EU FP COBWEB Project under grant no. 308513.

REFERENCES

Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., and Steggles, P. (1999). Towards a better understanding of context and context-awareness. In *Handheld and ubiquitous computing*. Springer.

Aghaei, S., Nematbakhsh, M. A., and Farsani, H. K. (2012). Evolution of the world wide web: From web 1.0 to web 4.0. *International Journal of Web & Semantic Technology*, 3(1).

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., et al. (2010). A view of cloud computing. In *Communications of the ACM*. ACM.

Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey. In *Computer networks*. Elsevier.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, 284(5):34–43.

Botts, M., Percivall, G., Reed, C., and Davidson, J. (2008). sensor web enablement: Overview and high level architecture. In *GeoSensor networks*. Springer.

Chi, E. H. (2008). The social web: Research and opportunities. *IEEE Computer*, 41(9):88–91.

Chu, X. and Buyya, R. (2007). Service oriented sensor web. In *Sensor Networks and Configuration*. Springer.

Delin, K. A. (2002). The sensor web: A macro-instrument for coordinated sensing. *Sensors*, 2(7):270–285.

Geihs, K. (2001). Middleware challenges ahead. *IEEE computer*, 34(6):24–31.

Grammel, L. and Storey, M.-A. (2010). A survey of mashup development environments. In *The smart internet*. Springer.

Greenfield, A. (2006). *Everyware: The Dawning Age of Ubiquitous Computing*. Peachpit Press.

Hadim, S. and Nader, M. (2006). Middleware: Middleware challenges and approaches for wireless sensor networks. *Distributed Systems Online*, 7(3):1–1.

Hendler, J. (2009). Web 3.0 emerging. *IEEE Computer*, 42(1):111–113.

Hendler, J. and Berners-Lee, T. (2010). From the semantic web to social machines: A research challenge for ai on the world wide web. *Artif. Intell.*, 174(2):156–161.

Issarny, V., Georgantas, N., Hachem, S., and Zarras (2011). Service-oriented middleware for the future internet: state of the art and research directions. In *Journal of Internet Services and Applications*. Springer.

Kambil, A. (2008). What is your web 5.0 strategy? In *Journal of business strategy*. Emerald Group Publishing Limited.

Marsh, D., Tynan, R., OKane, D., and P OHare, G. M. (2004). Autonomic wireless sensor networks. *Engineering Applications of Artificial Intelligence*, 17(7):741–748.

Milanovic, N. and Malek, M. (2004). Current solutions for web service composition. In *IEEE Internet Computing*. IEEE Educational Activities Department.

Mohamed, N. and Al-Jaroodi, J. (2011). A survey on service-oriented middleware for wireless sensor networks. *Service Oriented Computing and Applications*.

Murdoch, O. and Nixon, P. (2010). A unifying framework for situation identification methodologies. *MRC*.

O'Grady, M. J., Murdoch, O., Kroon, B., Lillis, D., Carr, D., Collier, R. W., and O'Hare, G. M. (2013). Pervasive sensing: Addressing the heterogeneity problem. In *Journal of Physics: Conference Series*, volume 450, page 012044. IOP Publishing.

O'Hare, G. M. P., Muldoon, C., O'Grady, M. J., Collier, R. W., Murdoch, O., and Carr, D. (2012). Sensor Web Interaction. *International Journal on Artificial Intelligence Tools*, 21(02):1240006.

Raychoudhury, V., Cao, J., Kumar, M., and Zhang, D. (2013). Middleware for pervasive computing: A survey. *Pervasive and Mobile Computing*, 9(2):177–200.

Rogers, Y. (2006). Moving on from weiser's vision of calm computing: Engaging ubicomp experiences. In *UbiComp 2006: Ubiquitous Computing*, pages 404–421. Springer.

Sadri, F. (2011). Ambient intelligence: A survey. *ACM Computing Surveys (CSUR)*, 43(4):36.

Tynan, R., Marsh, D., O'kane, D., and O'Hare, G. M. (2005). Intelligent agents for wireless sensor networks. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 1179–1180. ACM.

Wang, M.-M., Cao, J.-N., Li, J., and Dasi, S. K. (2008). Middleware for wireless sensor networks: A survey. *Journal of computer science and technology*, 23(3):305–326.

Weiser, M. (1991). The computer for the 21st century. *Scientific American*, 265(3):94–104.

Business Model Design
An Evaluation of Paper-based and Computer-Aided Canvases

Boris Fritscher and Yves Pigneur
Faculty of Business and Economics, University of Lausanne, 1015 Lausanne, Switzerland
boris.fritscher@unil.ch, yves.pigneur@unil.ch

Keywords: Business Model Canvas, Computer-Aided Business Model Design, Guidelines.

Abstract: In recent years, Business Model Canvas design has evolved from being a paper-based activity to one that involves the use of dedicated computer-aided business model design tools. We propose a set of guidelines to help design more coherent business models. When combined with functionalities offered by CAD tools, they show great potential to improve business model design as an ongoing activity. However, in order to create complex solutions, it is necessary to compare basic business model design tasks, using a CAD system over its paper-based counterpart. To this end, we carried out an experiment to measure user perceptions of both solutions. Performance was evaluated by applying our guidelines to both solutions and then carrying out a comparison of business model designs. Although CAD did not outperform paper-based design, the results are very encouraging for the future of computer-aided business model design.

1 INTRODUCTION

In a fast-evolving business landscape, companies need to turn to new methods to help them rethink their business strategy. By using a Business Model Canvas (BMC), they can get a better picture of their current business model, as well as create new ones. These methods are gaining in popularity, leading to the creation of a range of tools to support them. Thus, BMC design has evolved from being a paper-based activity to being one that is supported by custom-built computer-aided business model design (CABMD) tools. Such tools provide functionalities that are similar to the paper experience, but offer additional options such as version handling and calculation. However, in order to give free rein to creativity, the tools tend to be open in nature, making them difficult to use in a structured environment in which software tools are used. This is especially the case if the application is expected to assist the model itself. Guidelines can help by capturing and encapsulating knowledge that has been collected from best practice. This knowledge can then be offered to users. Elaborating guidelines helps in the design of more coherent business models; in turn, this helps to improve the way in which CAD can support business model design. Nonetheless, all these advanced CAD tools, which are aimed at supporting the BMC, are worthless if they hinder the creative-thinking process enabled by the paper version. However, if evaluation can show that a digital canvas is perceived and performs at least as well as a paper-based canvas, this promises great potential. For example, some features, such as automated guidelines validation, are only possible with digital tools.

The focus of this research can be summarized by the following questions:

Can guidelines help to produce a more coherent business model canvas?

How does using a computer-aided business model canvas design tool affect perception compared with using a paper-based version?

How does using a computer-aided business model canvas design tool affect performance compared with using a paper-based version?

In the next section we present any justificatory knowledge, followed by a short description of our methodology. We then present the guidelines themselves and the way in which they can be supported by CAD. Our evaluation also includes a business model case and an experiment aimed at comparing paper-based design with CAD. The results are presented, along with any lessons learned.

2 DESIGNING BUSINESS MODELS

According to Osterwalder and Pigneur (2013) there are three areas where IS research can contribute to strategic management. First, modelling at a strategic level requires a common language and representation. One business model visualization in particular is starting to be widely adopted by practitioners: the Business Model Canvas (BMC) (Osterwalder and Pigneur, 2010).

Second, the strategizing process should be seen as a design activity. Here, design means elicitation and testing; namely, the generation of ideas and their validation.

Third, they put forward the idea that CAD can *"make tasks easier and quicker, while revealing as-yet-unseen opportunities"* (Osterwalder and Pigneur, 2013).

For the purpose of our study, testing in business model terms represents two things: 1) coherence of the business model and 2) commercial viability of the business hypothesis. In this paper, we will focus on the former, since it can be addressed by CAD tools.

2.1 BMC Evaluation

The BMC design activity is usually a team effort that involves stakeholders across the company. A recent survey[1] of 1,172 users confirmed that 74% of them carry out design in groups of 2 to 10 collaborators. Moreover, from research undertaken by Reinig (2003: 65), we know that *"the satisfaction users have with the processes and outcomes of the teamwork itself often determines the ultimate adoption and sustained use of collaborative technologies"*. Therefore, it is important to compare users' perceptions of paper-based BMC with its computer-aided counterpart.

To date, few studies have sought to evaluate BMC design. However, Hoffmann et al. (2012) have shown that paper-based BMC design outperforms two other idea generation methods considerably. They noted that: *"The ability to select the best idea was found to be much higher when groups worked with the business model canvas: 80 per cent of groups selected the best idea"*. Their decision to limit their study to paper-based design was based on the extensive training and potentially expensive support systems required by electronic methods.

In their research, Lucassen et al. (2012) focused on how business model methods can be supported by software. They came to the conclusion that, *"BMC is the preferred method because it effectively models explicit information of both tangible and intangible aspects of the business and communicates this information in a highly accessible manner to parties unfamiliar with the modeling technique"*. However, they did point out that there is still room for improvement, because of a lack of clarity in the modeling process. Furthermore, they pointed out that knowing when the model is sufficiently correct is not explicit. This sustains the relevance of providing better business model design guidelines.

3 A DESIGN SCIENCE APPROACH

In this study, we used the methodology put forward in design science research by Gregor and Hevner (2013). First, we explored how CAD can best support business modeling. This was carried out iteratively by building and evaluating prototypes. We also focused on the evaluation of the perception and performance of CAD business model design in comparison with paper-based design. We used existing artifacts such as the BMC and CABMD tools. Our evaluation has one particularity in that we chose to use a commercial instantiation of CABMD software. However, we did propose a new artifact in the form of guidelines, with the intention of making better use of them. The evaluation of this artifact is done by validity. Demonstrating that a coherent business model case can be created by following the guidelines.

4 TOOLS FOR BUSINESS MODEL DESIGN

To help in the design of a BMC we put forward guidelines, aimed at helping both the elicitation of new elements and the testing of coherence. These guidelines could then be transformed into actionable rules for use inside a CABMD tool.

4.1 BMC as a Paper-based Artifact

The BMC uses nine building blocks to represent a business model. These building blocks can be further grouped into four perspectives, as shown in table 1. The main perspective is the offer (what we do), which connects the client perspective (who we do it for) and the activity perspective (how we do it).

[1] Internal survey, Business Model Foundry GmbH 2012

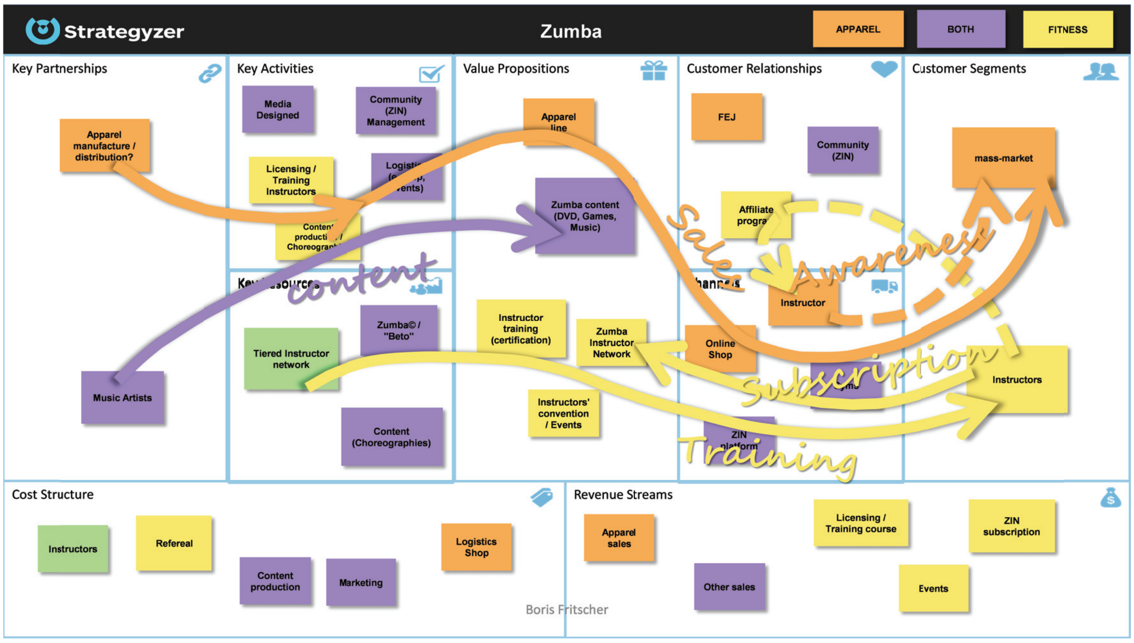


Figure 1: Business Model Canvas of Zumba Fitness.

Finally, the financial perspective deals with profit (how much?).

The positioning of these nine blocks is very important. Visually, they form separate groupings, which helps to structure the thought process and facilitate comparisons between the business models drawn using this method. As can be seen in figure 1, the offer is in the centre; to the right is the client perspective and revenue stream, whilst to the left is the activity perspective and cost structure.

Table 1 Business Model Canvas Components.

Perspective	**Question**	**Building block**
Offer	What?	Value proposition
Client (right side)	Who?	Customer segment Distribution channels Customer relationships
Activity (left side)	How?	Key resources Key activities Key partnerships
Financial	How much?	Revenue stream Cost structure

There are three guidelines which help in the elicitation of business elements on the BMC.

A. Discover business model elements

Any elicitation technique is applicable. Many users go through the nine blocks one after another and add elements as if it were a checklist. However, this does not harness the full potential of the model, because its strength lies in the connectedness of elements from different blocks.

B. Improve business model through connections

A good BMC has all of its elements connected to at least one other; there are no orphan elements. Additionally, the number of elements inside each block has to be reasonable so as not to overload the visual appearance. This is accomplished by displaying only those elements that perform an essential role in the business model. Furthermore, the connectedness between elements helps in telling the story of the business model.

C. Highlight business model mechanics

By using big arrows on the top of the BMC to depict the flow of interaction, it is possible to visualize the story of the business model. In section 5 we use an example to illustrate the three guidelines.

4.2 BMC Coherence Guidelines

Testing a BMC's coherence involves the verification of control points on three levels: elements, building blocks and connections. We propose a set of guidelines for each.

4.2.1 Guidelines Applying to Any Individual Element

These guidelines help in maintaining a visually understandable BMC.

- There is only one idea per sticky note.

- Ideas are written with keywords, or presented with a simple illustration.
- The meaning of the element is understandable by all stakeholders.
- The element is a key component in explaining the business model; indeed, without it the business model cannot be explained.

4.2.2 Guidelines Applying to Individual Building Blocks

These guidelines help to identify the right amount of detail for the BMC.

- All nine building blocks of the model are used, or have at least been considered.
- Elements that are too detailed have been grouped into a simpler element.
- Elements that are too generic have been split into more detailed elements.
- The detail level of the elements are adequate (there are not too many detailed elements, nor to few which are too generic).

4.2.3 Guidelines Applying to Connections between Elements in Different Building Blocks

These guidelines help with the coherence of the BMC.

- Colors are used on elements to highlight their connections according to the BMC's meta-model (Fritscher and Pigneur, 2010)
- Each color is labeled and has a specific meaning.
- Client perspective is valid:
    - Each customer segment is addressed by one or more value proposition.
    - A channel supports a value proposition-customer segment set.
    - If present, a customer relationship targets a customer segment.
    - In case of multiple customer segments, colors distinguish each business side.
- Activity perspective is valid:
    - Each value proposition is produced/delivered by a key activity, a key partner or offers a key resource.
    - Key resources or key partners support an activity.
- Financial perspective is valid:
    - Revenue stream is generated from a value proposition-customer segment set. (A revenue stream can also be "free").
    - Major fixed costs are listed.
    - Major variables costs are listed.
- There are no orphan elements: all elements are connected to another element (in a different block to themselves).

4.3 BMC Computer-Aided Design

Multiple versions of BMC prototypes can be found, as well as commercial versions. Research prototypes emphasize advanced features; however, they lack finesse in user experience. In order to make the best comparison between a paper-based BMC and a digital implementation, we chose to use Strategyzer, a commercial version that is closest to the original paper-based BMC. This commercial software solution not only has a proven user-friendly interface, it has the added advantage of being inspired by the same original artifact ideas as our research prototypes. Another benefit is that it has calculation features which sit on top of the basic functionality features, showing that integration is possible without compromising the simplicity of the user interface.

When Computer Aided Design (CAD) is applied to the BMC, it can support elicitation by making it easier to move, duplicate and rename elements. Thanks to its digital properties, elements can also be hidden and shown selectively, allowing for multiple views of the same data. This enables the exploration of business model variants, thereby further aiding the elicitation process.

Beyond visual interactions, such software tools can be used to support business model design with features that are tailored to guarantee the coherence of the meta-model on which they are built. Guidelines can be transformed into rules, which can then be tested by the tool. In case of incoherence, a notification is shown on any invalid elements. Such visual flags can, in addition, contain hints on how to fix the problem or, at the very least, offer a reference as to which rule or guideline was violated. The computation is carried out automatically; thus, visual flags appear as soon as something changes.

Guidelines allow a coherence score to be attributed to each model; this score is based on the number of fulfilled conditions.

4.3.1 Example Guideline Transformed into Rules and Resolution Hints

Rule 1: There are not more than a specified number (given by a threshold) of customer segment elements with the same color.

Resolution Hint: Either merge elements that are too detailed (building block guideline) or change colors of element belonging to a different value proposition to distinguish the segments (connection guideline).
Rule 2: A customer segment has to have a corresponding value proposition element with the same color as itself.
Resolution Hint: Create missing elements or add right colors.

5 COMPARING COMPUTER-AIDED DESIGN WITH PAPER-BASED DESIGN

The focus of our evaluation is to compare a paper-based BMC with one created using a computer-aided design tool in terms of perception and performance. In this section, we first present how we created a business model for Zumba Fitness following our guidelines. We then go on to present the experimental setup, followed by the results and statistics.

5.1 Zumba Fitness Business Model

This case is used to illustrate how to apply our elicitation guidelines when designing the Zumba Fitness business model using publicly available information (as shown in figure 1). Zumba Fitness is a company that offers fitness training to instructors (yellow) and sells fitness apparel (orange) to the mass market. Separate colors were used for each type of offering. Elements that are affected by both value propositions are shown in violet.

A. Discover Business Model Elements

The discovery of elements, which can be added to any of the building blocks, can come from internal knowledge, interview, observation or indeed any kind of research method. However, it is crucial to move from one idea to the next without limiting oneself to one block at a time. Our main source of information for this case study was a six-page report by Inc magazine[2] and a video interview featuring one of the company's founders.

As should be the case for any presentation of the BMC, we will first present the elements as a story, instead of going through the blocks one at a time.

Zumba Fitness offers *Instructor training* to the *instructors* customer segment with the help of their *online ZIN platform* and *gyms*. Giving courses generates *licensing/training* revenues. A second revenue stream from *instructors* is a *subscription* to the *ZIN network*. This offer (value proposition) gives the *instructors* access to new *Zumba content* which they can use in their own Zumba classes. To provide the aforementioned value propositions, a number of key activities have to be performed, including *training*, *ZIN community management* and *creation of new content* (choreography).

Another customer segment is the *mass market*, namely, people who buy apparel from the *online shop*, thus generating *sales* revenue.

B. Improve Business Model through Connections

With any BMC, it is important to check the connections between the elements. This helps to identify any missing elements. It can also lead us to question the validity of elements if no connection to other elements can be found.

Continuing with our example, although fitness apparel is sold, its source is missing. Therefore, for coherence, *manufacturing & distribution* partners had to be added, as well as a *logistics* and *media design* activity, and the cost structure of a *logistics shop*.

The content creation activity produces new *choreographies*, not only as a value proposition, but also as a new resource. However, to produce such choreographies, the company also needed *music artists*; these become a new partner element. The creation of content (*content production*) is also a major cost in the business model. An additional resource, which gives value to their content, is the *Zumba* brand name.

C. Highlight Business Model Mechanics

Business Model mechanics help to visually illustrate major interactions between elements on the BMC. The flow of the interaction is depicted by large arrows, which connect the elements. Thinking about the mechanics and the story behind it will help reveal additional element interaction, which may not emerge when looking only at individual elements.

This case is particularly interesting, because a series of mechanics helps to reveal that instructors are also a channel. Zumba starts by *training/certifying* instructors; a major percentage of these instructors will then subscribe to the Zumba Instructor Network. A certified instructor goes on to give Zumba courses and naturally starts to promote

[2] http://www.inc.com/magazine/201212/leigh-buchanan/zumba-fitness-company-of-the-year-2012.html

Table 2: Descriptive statistics.

Variable	Mean		Std. Dev.		Min		Max	
Tool	P	S	P	S	P	S	P	S
Perceived Usefulness	2.38	1.83	0.98	0.69	1.25	1.00	5.00	3.50
Perceived Ease of Use	2.36	1.80	1.13	0.73	1.00	1.00	6.00	4.00
Task Outcome	2.27	1.85	0.80	0.72	1.00	1.00	4.33	3.33
Task Innovation	2.17	2.79	0.73	1.10	1.00	1.00	3.00	4.67
Total Elements	23.00	28.82	5.75	6.91	11.00	20.00	32.00	42.00
Correct Elements	16.81	15.64	2.62	3.91	13.00	11.00	20.00	25.00

P: Paper (21 observations), S: Strategyzer software (22 observations)

Table 3: Pearson's correlation between concepts.

	PU	PEU	TO	TI	TE	CE
Perceived Usefulness [PU]	1.00					
Perceived Ease of Use [PEU]	0.54***	1.00				
Perceived Task Outcome [TO]	0.27		1.00			
Perceived Task Innovation [TI]			0.39**	1.00		
Total Elements [TE]	-0.28	-0.26			1.00	
Correct Elements [CE]				-0.32*	0.43**	1.0

P>|t| *** 0.001, ** 0.01, * 0.05

the brand and its apparel. To build on this phenomenon, Zumba offers them an *affiliate program* (customer relationship). Thus, through awareness generation, *instructors* become a channel to the mass market. This supports the second mechanic, which is the *sale of fitness apparel* through the *online shop*. The third mechanic can be found backstage, in the form of generating content.

Having added *instructor* as a channel and an *affiliate program*, it is then necessary to check again for any connections. In turn, this reveals that, to retain coherence, a *referral fee* has to be added to the cost structure. This demonstrates the need to iterate through the mentioned techniques and guidelines until everything is in a stable and coherent state.

5.2 Experiment Setup

Our experiment was aimed at designing a business model for the Zumba case using an article and a video interview as information resources. The evaluation was performed during a business model course attended by students from a master's program in IS. The students were all familiar to a similar level with the BMC method and web tool. A total of 43 students participated. They were split into 22 groups in order to have the most groups possible and avoid students having to do the task individually. Having teams of two people is a key component of generating creative ideas (Paulus 2000) and corresponds better to the normal use of the BMC.

Half of the groups were asked to do the design task using a paper-based BMC. The others used the chosen computer-aided business model design software and were not allowed to use paper at all.

Evaluation of the task was carried out in two parts. First, when a group considered their work to be complete, each student was asked to individually fill out a questionnaire to assess their perception of the task. Second, all BMCs were collected and evaluated to assess the performance of the groups' designs.

5.3 Results and Statistics

In this section, we present our measures of perception and performance, followed by their statistical analysis.

5.3.1 Measurement of Perception

For the questionnaire we decided to use questions and scales taken from existing literature (see appendix for the full question list). The concepts of perceived usefulness and perceived ease of use were adopted from TAM (Davis 1989). However, we simplified the questionnaire, reducing the number of questions by removing those with similar meanings, to avoid confusion amongst our non-native English speakers.

A measure of the perceived task outcome was added so that we could test whether there is a difference in perception between the two medias. In

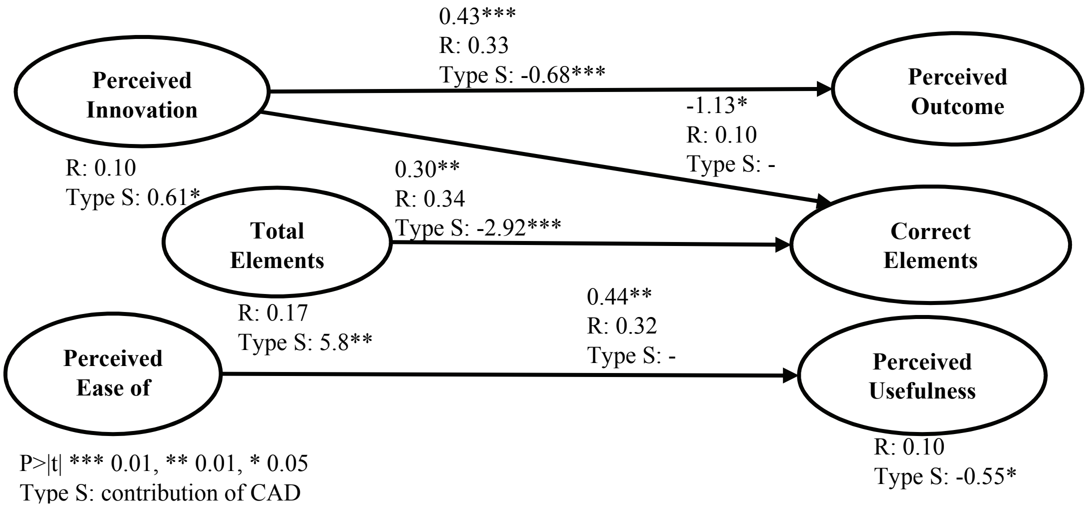


Figure 2: regressions between concepts.

addition, this allowed us to make a comparison with the real outcome performance metric. The task outcome was adapted from Briggs et al. (2006). Here, we selected items from their meeting outcome and meeting process questions.

We were also interested in how media type impacts our perception of being able to generate ideas. For task innovation we used questions taken from (Torkzadeh and Doll, 1999).

All answers have a seven-point likert scale, which we coded from 1 (best) to 7 (worst).

In order to analyse the concept, for each question we grouped the answer variables of each metric into a usable concept (latent variables) using Cronbach's alpha. The *perceived usefulness* concept is well defined by its four questions with an alpha of 0.75. For the *perceived ease of use*, we dropped question number 2.3 to get a better alpha of 0.71. For the *task outcome* concept we had to drop question 4.3 to get an acceptable alpha of 0.75. The *task innovation* concept is well described by its three questions with an alpha of 0.92.

5.3.2 Measurement of Performance

The designed business model's performance was computed by comparing it with the solution developed by two experts who followed the techniques and guidelines presented in the artifact section. A total of 28 points could be achieved for the *Correct Element* measure. The comparison points were not all a direct match; if an element was similar in meaning to the solution, it was also accepted. There were no negative points for additional elements and the same evaluator corrected all of the BMCs. We also took into consideration the metric of the *Total Elements* in order to measure any differences in quantity generation between the media.

5.3.3 Descriptive Statistics of Results

As can be seen in table 2, answers are skewed positively, with a low average score for all the perception constructs. This indicates that overall the students had a very positive perception of the BMC, irrespective of the type. The computer-aided canvas was marginally better than the paper-based canvas on all the perception measures, except for task innovation. It also helped to generate more elements. Correct elements are very similar for both types. Element metrics of the computer-aided canvas showed the greatest deviation, with both the best and the worst number of correct elements.

5.3.4 Statistical Analysis of Concepts

We used the Stata 12 software package to perform our statistical analysis. After verifying the concept's alpha values we looked at the Pearson correlation between them. The matrix, which can be seen in table 3, helped us select the concepts that warranted further analysis with regressions to determine the impact of the type of media used.

The strongest correlation is between usefulness and ease of use, which matches TAM's theory. The correlation between the total elements and correct elements also seems natural. We did not penalize wrong elements, therefore the more there are, the greater the possibility of also having correct ones. Of particular interest is the correlation between task innovation and task outcome, and between task innovation and correct elements, which represents the real outcome. We explore these relations further in the discussion.

6 LESSONS FROM THE COMPARISON

A regression analysis was used on the variables for which correlations stood out. The results are shown in figure 2. Only links with significant regression results are shown. Type S is the contribution of using the computer-aided software BMC over the paper-based BMC. As already observed, with the mean values, *perceived innovation* is slightly better with the paper-based BMC, but the R-square value is only 0.10. On the other hand, *perceived innovation* strongly predicts *perceived outcome*. Users of the digital BMC perceived that it helped them do a better job more than did the users of the paper-based BMC. *Perceived innovation* slightly predicts *real outcome* (correct elements), without a difference between types.

On its own, *perceived usefulness* is seen as being better with the digital tool. This could be a bias of the population of IS students who are familiar with IT technology and might prefer a technical solution to one that uses paper.

There is no significant difference between the type that affected the influence of *perceived ease of use* over *perceived usefulness*. This can be seen as a positive result for the software tools, because it does not perform better or worse. Having at least the same ease of use as paper is a key result, which should be reflected upon when considering that the digital tool has the potential of offering additional features, providing usefulness that is not possible on paper.

The computer-aided BMC helps to generate more elements than a paper-based one; however, it also has a negative influence on the number of correct elements. It is easier to generate more elements, but also to generate more wrong elements.

Users who think that the digital tool helps them innovate, think they have performed better; however, in our small setup they obtained similar numbers of correct elements.

In addition to the statistical analysis, we also observed how the teams worked during the design task. One observation that is of particular interest relates to the process of eliciting elements. On the paper-based BMC, a discussion first occurs and then a sticky note element is created and positioned. On the computer-aided BMC, however, which also supports collaboration, elements are added first by each member and then changed to reflect the consensus. This is interesting because recording the decision inside the tool means that it can be utilized to better support the ongoing business modeling collaboration process.

Three weeks after the first task, we carried out a trial experiment with the coherence guidelines using paper. The results were varied and inconclusive, although users did say it helped them improve their model. Problems arose when attempting to test them on paper. In this situation, users have to perform the checks manually; in some instances, they do not take the time to iteratively do it as soon as they change something. Therefore we posit that although we showed that guidelines can be used to create coherent models on paper, it is more appropriate for such guidelines to be implemented and tested inside a prototype tool. Here, they can be recomputed each time a change is detected.

In summary, in our experiment with our test group, the tested CAD tool was as effective as paper-based design for the creation of business models in terms of eliciting elements of the BMC. This indicates that with the help of rules, it might be better suited for testing the coherence of business models than paper-based design.

7 CONCLUSIONS

To assist BMC design using software tools, we proposed guidelines that help with elicitation and testing in order to produce coherent models. Before implementing such features in a digital tool we needed to confirm that perception and performance on a basic BMC design task are at least similar to those of a paper-based design. With our evaluation we found that the tested digital tools can be perceived as useful, and does not perform any worse than its paper-based alternative. Even if CABMD did not outperform paper-based design, it shows some promising results, because such tools can be extended to offer additional features, thus increasing their usefulness. Features that are much better suited for digital tools include the continuous reviewing of coherence rules to check their validity.

In this paper, we focused on modeling an existing *"as-is"*, business model. Further research is needed to explore options that may enable the exploration of future *"to-be"*, business models. For example, rules could be extended to simulate financial assumption or validate regulatory constraints.

REFERENCES

Briggs, R. O., Reinig, B. A., & de Vreede, G. J. (2006). Meeting Satisfaction for Technology-Supported Groups an Empirical Validation of a Goal-Attainment Model. Small Group Research, 37(6), 585-611.

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS quarterly, 319-340.

Fritscher, B., & Pigneur, Y. (2010). Supporting Business Model Modelling: A Compromise between Creativity and Constraints. Task Models and Diagrams for User Interface, (LNCS 5963), 28–43.

Gregor, S., & Hevner, A. R. (2013). Positioning and Presenting Design Science Research for Maximum Impact. MIS Quarterly, 37(2).

Hoffmann, F., Eppler, M. J., & Bresciani, S. (2012). Business Model Ideation: An Experimental Approach for the Evaluation of Team-based Methods. 12th EURAM 2012 European School of Management

Lucassen, G., Brinkkemper, S., Jansen, S., & Handoyo, E. (2012). Comparison of visual business modeling techniques for software companies. In Software Business (pp. 79-93). Springer Berlin Heidelberg.

Osterwalder, A., & Pigneur, Y. (2010). Business model generation: a handbook for visionaries, game changers, and challengers. John Wiley & Sons.

Osterwalder, A., & Pigneur, Y. (2013). Designing Business Models and Similar Strategic Objects: The Contribution of IS. Journal of the Association for Information Systems, 14.

Paulus, P. (2000). Groups, Teams, and Creativity: The Creative Potential of Idea-generating Groups. Applied psychology, 49(2), 237-262.

Reinig, B. A. (2003). Toward an understanding of satisfaction with the process and outcomes of teamwork. Journal of Management Information Systems, 19(4), 65-84.

Torkzadeh, G., & Doll, W. J. (1999). The development of a tool for measuring the perceived impact of information technology on work. Omega, 27(3), 327-339.

APPENDIX

The following questionnaire was used for our survey, either using *Strategyzer* or *the paper canvas* as subject.

Based on your short experience with Strategyzer, how would you rate the following statements when thinking about using Strategyzer for future Business Model Design Tasks?

The following seven point Likert scale was used: extremely likely (1), quite likely (2), slightly likely (3), neither (4), slightly unlikely (5), quite unlikely (6) extremely unlikely (7)

1 Perceived Usefulness

1.1	Using Strategyzer to design business model would enable me to accomplish the task more quickly.
1.2	Using Strategyzer would improve my performance in designing business models.
1.3	Using Strategyer would make it easier to design business models.
1.4	I would find Strategyzer useful for designing business models.

2 Perceived Ease of Use

2.1	Learning to operate Strategyzer to design business models would be easy for me.
2.2	I would find it easy to get Strategyzer to do what I want it to do.
2.3	It would be easy for me to become skillful at using Strategyzer to design business models.

The following seven point Likert scale was used for the next two sections: strongly agree (1), agree somewhat (2), agree (3), neither (4), somewhat disagree (5), disagree (6), strongly disagree (7)

Now evaluate your business model design task.

3 Task Outcome

3.1	I feel satisfied with the designed business model.
3.2	I feel satisfied with the process used to design the business model.
3.3	With more time I could substantially improve the designed business model.
3.4	I had enough time to complete the task.

4 Task Innovation

4.1	Strategyzer helps me create new ideas.
4.2	Strategyzer helps me come up with new ideas.
4.3	Strategyzer helps me try out innovative ideas.

BYOD: The Next Wave of Consumerization of IT
The Impact of BYOD on the Enterprise IT Landscape

Ivan I. Ivanov
Empire State College of the State University of New York,
Long Island Center, Hauppauge, NY, U.S.A.
ivan.ivanov@esc.edu

Keywords: Bring Your Own Device – BYOD, Consumerization of IT, Company-Owned, Personally-Enabled – COPE, Mobile Device Management – MDM, Mobile Application Management – MAM, Mobile Content Management – MCM, BYOD Framework

Abstract: In the last few years, new technologies emerge first in the consumer market and then, after mass acceptance, are employed largely by business organizations. Companies across the globe are going through the most disruptive new technology development: Consumerization. Consumerization of IT, along with workforce mobility, and flexible, reliable, accessible and affordable remote computing, are forcefully changing the corporate IT lanscape, affecting the relationship between enterprise IT, knowledge workers, corporate users, and consumers. This phenomenon advances with every arrival of new devices, applications, or strategic trends such as Bring-Your-Own-Device (BYOD). BYOD is currently a growing trend in the private and public sector that allows employees the convenience of logging into the corporate network with their personal mobile devices. This paper explores the impact of this trend on the enterprise IT landscape and provides a decision framework for BYOD adoption.

1 CONSUMERIZATION OF IT – THE AGE OF THE CUSTOMER

Current trends in IT utilization show that new technologies emerge first in the consumer market and then, after mass acceptance, are employed largely by business organizations. The expected consequence of this pattern is that across the globe companies are experiencing the most disruptive new technology trend of this decade: Consumerization.

The process of Consumerization is well depicted by Forrester Research report from June 6, 2011, "Competitive Strategy in The Age Of The Customer." This is the phenomenon of employees using devices, applications, and web services to actually empower business users or employees to innovate (Forrester Consulting, 2013).

Consumerization of the IT is actually advancing swiftly with every arrival of new devices, strategic trend, or applications. The oportunities for business users, ranging from consultants, hightech professionals and executives, to administrative assistants, and sales and call center representatives, to leverage consumerized offerings – those offered outside of organizational IT – varies depending on the business policy and openness.

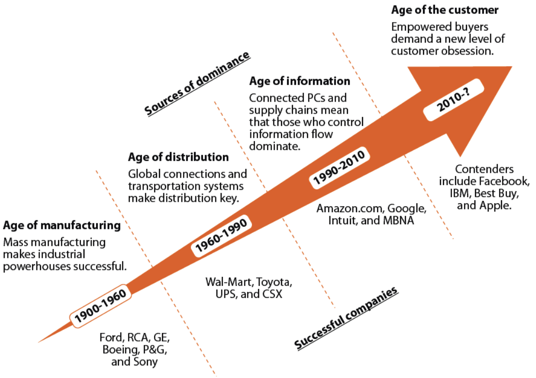

Age of the customer
Empowered buyers demand a new level of customer obsession.
Sources of dominance
Age of information
Connected PCs and supply chains mean that those who control information flow dominate.
2010-?
Age of distribution
Global connections and transportation systems make distribution key.
1990-2010
Contenders include Facebook, IBM, Best Buy, and Apple.
Amazon.com, Google, Intuit, and MBNA
Age of manufacturing
Mass manufacturing makes industrial powerhouses successful.
1960-1990
Wal-Mart, Toyota, UPS, and CSX
1900-1960
Successful companies
Ford, RCA, GE, Boeing, P&G, and Sony

Figure 1: Consumerization of IT. Adopted from "Competitive Strategy in The Age Of The Customer," Forrester Research, Inc. June 6, 2011.

Consumerization of IT, along with workforce mobility, and flexible, reliable, accessible and affordable remote computing, are forcefully changing the corporate IT lanscape affecting the relationship between enterprise IT, knowledge workers, corporate users, and consumers. For company IT management, consumerization exemplifies the convergence of a demanding set of challenges such as information and infrastructure security, technology policy, data protection, and end-user technology. For corporate management, consumerization of IT signifies a new strategy which supports business models and process innovations, talent strategy and customers' satisfaction, as well as corporate brand and identity.

Consumerization of IT blurs the line between personal and work life, especially for mobile workers. Mobile workers make up about 39% of the employees in Noth America, 25% in Europe, and 42% in Asia, with a growing tendency according to Forrester's analysis (Forrester Consulting, 2013). Their cohort benefits the business immensely by increasing productivity, and advancing collaboration and business agility, thereby improving customer satisfaction and climbing the rate of talent retention. Consumerized employees spread the boundaries of the workday and workplace, and it is fair to name them "anytime, anywhere workers."

2 BYOD – OPPORTUNITIES, CHALLENGES, GAINS

BYOD, or Bring-Your-Own-Device, is a growing trend in the private and public sector that allows employees the convenience of logging into the corporate network with their own personal devices. The rise of the mobile workforce to 1.2 billion in 2013, representing 35% of the worldwide workforce according to IDC Forecast, drives strongly the BYOD initiative as many of those workers will be using their own devices (International Data Corporation IDC, 2012).

The BYOD trend, as it is driven mostly by current Consumerization of IT in the enterpise, is forcing companies to redesign or create new policy and rules on how smart portable devices can be used for both corporate and private purposes, and how the related expenditures should be covered.

2.1 BYOD as Mobile Workforce Advancement

BYOD can be facilitated through applications that are native to the device, downloaded or installable applications, or even a mobile web browser. The BYOD boom originated via two converging trends: the need for employees to be responsive in a global, always-connected world, and the desire to save money by not replicating a device that employees may already own.

BYOD eliminates time boundaries, allowing employees to be productive during and after working hours. It offers flexibility, allowing them the ability to be connected to the corporate network and do some work, for example, during a child's baseball game in the afternoon as well as later in the evening. While BYOD offers flexibility and efficiencies, the initiative also brings a significant advantage to overall productivity through timely reactions and collaborative interactions. Mobility obviously brings significant improvement and added opportunities to the business environment, but as with many advantages, there are consecutive tradeoffs.

2.2 BYOD – Challenges and Concerns

A considerable challenge for an IT department is how it can effectively secure and manage the corporate network and information systems access for user-owned devices. For example, these devices cannot easily be identified, and therefore cannot be managed by the traditional IT department security settings. When employees bring in their own devices, IT loses significant control as it does not know where the device has been or what applications the user has downloaded, or what device has been introduced into the network.

Apart from legal and ethical issues, some of the noteworthy technical concerns of a BYOD program are as follows:

- delivering secure, remote access for mobile devices, while continuing to enforce granular access controls on network resources,
- securing corporate and personal data and mobile devices from malware, viruses and malicious applications, and
- mitigating the risk of loss, theft or exploitation of corporate and personal data residing on mobile devices.

From an ethical and legal standpoint, BYOD issues include corporate guidelines of what is considered "acceptable use" for actions an employee can take on their own device. Several laws have already been released regarding BYOD which include having employees sign an acceptable use agreement stating that the device can be seized for an indeterminate amount of time if the data on it is part of a legal dispute. Employees may have a real issue with their employers dictating to them what they can and cannot do on their own personal devices. It is imperative that businesses, schools, and other organizations establish a system of accountability if they allow a BYOD policy.

Privacy concerns are at the top of the list when it comes to legal issues regarding BYOD initiatives. Convergence of both personal and corporate data and applications presents a complex issue. Will the employer be permitted to access an employee's own e-mails and text messages on a personal smartphone or tablet used by that employee at work? Other considerations include access to browser history and installed software. This is a multifaceted problem that companies have to strategize prior to the adoption of a BYOD policy. Along with an acceptable use policy, many companies also educate employees as to what constitutes acceptable use. Adherence to the policy is only effective if there has been proper education for employees and whoever else has access to corporate information.

Some companies, such as IBM, instituted policies that banned its 400,000 employees from using two popular consumer applications over concerns about data security. The company banned cloud storage service Dropbox, as well as Apple's personal assistant for the iPhone, Siri. Siri listens to spoken requests and sends the queries to Apple's servers where they are deciphered into text. Siri can also create text messages and emails on voice command, but some of these messages could contain sensitive, proprietary information.

2.3 The Business Sense of BYOD

While BYOD offers flexibility and efficiencies, the initiative brings a significant advantage to productivity. The quarterly Mobile Workforce Report from iPass Company found that many employees are working up to 20 additional hours unpaid as a result of the companies' BYOD policy (ComputerworldUK, 2012).

The 2013 iPass/MobileIron Mobile Enterprise Report depicts the tendency of increasing usage of employee-owned smart devices from 42% in 2011 to 47% in 2012, while the percentage of smartphones provisioned by employers declined from 58% to 49% for the same period (Appcelerator, Inc., 2013). Executives actually stimulate the process of establishing corporate guidelines and policies to foster BYOD adoption. According to the same report, 56% of the IT managers in 2012 confirmed that their IT policies had become more responsive and flexible to employees' demands of utilizing personal devices for dual corporate and private usage.

3 BYOD – STRATEGY AND GOVERNANCE

At this time, for many organizations, BYOD has remained an informal practice and an escalating IT complexity because of this mess-up exposes those institutions to risks from security and compliance gaps. BYOD policy can vary substantially for different organizations depending on their priorities, industry regulations, or operational models. A successful BYOD program would combine effective infrastructure and data security with easy personal use.

3.1 BYOD and Mobile Management Solutions

The IT department should provide users with secure access to corporate applications and data, while adding role-based access control and security settings of personally-owned devices to prevent the organization against data loss and non-compliant usage. The lack of standartization reflects the diversity of devices and operating systems that IT departments must grapple with days, and actually creates market opportunity for mobile management providers (iPass Inc., 2013).

3.1.1 Mobile Device Management

An important component in deploying a BYOD program is the ability to manage the mobile devices that would interface with the enterprise network. Mobile device management (MDM) is the software solution that allows a network administrator to manage and control mobile devices such as smartphones and tablets. Ideally, the MDM should be able to interface with all types of mobile devices, operating systems, and the apps they run. Another consideration is the MDM must be able to operate with a number of providers used by employees. Typically in a corporate landscape, business-owned devices will use only one wireless service provider, but in a BYOD environment, employee-owned devices each have their own wireless providers and as a result, the MDM system will have to work with dozens of service providers.

The main purpose of an MDM system is to optimize the functionality and capabilities of the workforce mobile device while keeping the business IT infrastructure and data secure. Network administration in an MDM includes the ability to interface with the enterprise's current servers and systems so that it can manage and secure corporate-owned data and applications; synchronize with the mobile devices for file sharing, sending out patches, and add and remove devices from the network, and all these tasks must be able to be done directly over carrier networks.

By controlling and protecting the data and configuration settings for all mobile devices in the network, MDM should be able to prevent network and system exposure from a variety of threats. One of the leading mobile device management solutions is the MobileIron MDM system. This MDM software is part of the MobileIron IT Platform which provides a "turnkey" ability to have the mobile device user to interface with the corporate backend just by downloading an app. Once the device is authenticated and activated in the network, the user has access to all resources and content they have permission to access. Like with IT networks, mobile device management is one layer of a secure BYOD program. Device management is just the beginning and is the foundation to mobile application management (MAM) and mobile content management (MCM) implementation – the two other services needed to support BYOD initiatives.

3.1.2 Mobile Application Management

As MDM focuses on the management of mobile devices, mobile application management (MAM) focuses on the management of the applications used by mobile workers. This tool allows system managers to monitor, provision, install and uninstall, update, and audit software programs and applications for mobile devices. MAM functions similarly to network system management tools used within a network environment, but it is designed specifically to work with the unique characteristics of a mobile device. Management of a variety mobile devices in a BYOD requires the ability to manage different operating systems and integrate with the wireless service providers used by the employee. Another issue unique to mobile devices application management is that apps typically are installed using the pull method initiated by the user and not the push method. Additionally, some apps require payment before installation.

MAM addresses these two issues by allowing managers to create a catalog or app storefront of internally developed business apps, as well as making available license files or tokens for approved public apps so there is no need for users to pre-pay prior to installation. For example, Mobile Application Distribution Library is a customizable app distribution tool offered by MobileIron. The library makes public and private apps available to its end users to install on their mobile devices. In addition, internally developed apps can be made available to users without having to publish them in a storefront such as Apple App Store or Android Market (MobileIron, 2014).

Security is another function of the MAM. Mobile workers will access sensitive data and the backend systems using a variety of apps. It is imperative that the apps comply with organization policies and are properly validated and encrypted. As mobile device computing grows, the number of apps has increased and malicious apps have become as ubiquitous as the safe apps. The highest priority for MAM is to protect data that moves throughout mobile workflows as well as protect its backend system from these types of apps. Some of the approaches of MAM are to allow managers to configure application settings, profiles and credentials for enterprise authentication. MAMs also monitor application usage by observing traffic and application connections. These observations may generate reports and logs that can identify issues within the network and mobile device connections.

Mobile App Containerization is one of the most advanced MAM solutions offered by Good Technology that actually does not require a mobile device management system. Containerization protects data transfer through the use of strong separation of personal and business apps and data. Containerized apps ensure that the enterprise security protocols and encryption throughout the transmission remain constant and consistent until the employees have completed their task. In addition, this method allows the manager to wipe company-owned data and apps only from a lost or stolen device; personal data is not touched at all. This actually addresses a major BYOD issue of how to manage business work product without breaching private personal data and information.

3.1.3 Mobile Content Management

Mobile content management (MCM) is the third layer that is needed to manage a BYOD enterprise level environment. Content is where users are very much hands on. For corporate managers, it is important that the content users access is in an environment that is secure and accessible, while offering the ability to share and collaborate. Mobile workers not only want but need to access the most current business documents and content quickly, anytime and anywhere, without worrying about security risks. This security employs a variety of authorization and access permissions. These may include user authentication by logging-in to the system as well as entering authentication codes for particular documents.

Since mobile workers are not necessarily located at a fixed site, content may be dynamic depending on the worker's location through the use of a global positioning system (GPS) or navigation system. AirWatch's MCM application is called AirWatch Secure Content Locker; this is a secure centralized storage for all business content, files, and documents with three storage options. The cloud storage option can integrate with the most popular repositories such as Google Drive, Office 365, SkyDrive, and Amazon EC2. The on-premise option can integrate with the host repositories as well as a secure access to SharePoint without a VPN connection (AirWatch, 2013). The third is the hybrid option of the cloud and on-premise. AirWatch's MCM also facilitates two-way synchronization of content from users' desktops to mobile devices. So individual, user-created content can be accessible to any device that user owns. This option is exactly what the mobile worker needs and expects when working within a mobile-first environment. Again, the expectation is that these documents are always available with any device that is used to call it.

3.2 BYOD – A Decision Framework

The serious challenges in developing a BYOD strategy and the consecutive framework for its implementation is the impact BYOD can have on individuals' privacy, organizational security, and the liability of both entities. The Gartner analysts Andy and Nick Jones in their Checklist for Determining Enterprise Readiness to Support Employee-Owned Devices have analysed and defined a structured approach in seven phases on the road to this emerging trend (Rowsell-Jones, 2012).

For corporate IT structures embracing a BYOD framework, the following key steps should be considered:

- Reasoning and deciding on a BYOD strategy – identify corporate mobile needs; define BYOD scope; shape sponsors' and stakeholders' commitments and responses to a BYOD program
- Design BYOD program segmentation by roles/needs/functions in the organization – categorize internal and external support, the range and type of access, and create packages of Policies and Technologies for each group
- Plan BYOD implementation by streamlining tools and technologies, network infrastructure and services, financing models, and exit options such as:
  - classify and approve list of devices and versions of mobile operating systems, applications, and providers;
  - design uniform policies, to enable scalable control and management of the user-owned mobile device utilizing Mobile Device Management (MDM), Mobile Applications Management (MAM), and Mobile Content/Document Management (MCM) solutions;
  - acceptable use policy with user's responsibilities and organization's rights against user's possession;
  - reimbursement plan options, total cost of ownership, corporate/private ownership separation, and list of approved exit options.
- Program setup and approval – complete internal policy, procedures, contracts, agreements, and training documents; educate stakeholders and ensure their sign-off; gain sponsors' budget and program approval
- Perform proof of concept by running a pilot over selected BYOD segmentations – modify procedure/policy/technologies based on the feedback and lessons learned from the pilot
- Program execution and evolution – periodic review and update of the BYOD program with current software versions, devices, applications, and providers. Utilize and evolve a mobile systems' features/limitations framework that supports the adopted corporate mobile management system(s).

The early BYOD adoptions have already experienced numerous concerns regarding losing personal data and privacy as corporations took full control over personal devices, applications, and information by utilizing mobile device management and device-level layer 3 VPNs. To address most of those critical anxieties, instead of a full control of the personal device, most corporations currently focus on adopting a set of tools to enable IT departments to wrap corporate applications in a security layer and to make sure that the enterprise control on the personally-owned device is limited only to the corporate data and applications. This actually shifts from MDM to MAM and from device-level VPNs to explicit application-specific VPNs involving technologies such as BIG-IP, APM, AppTunnels and encrypted connection to specific service supported by Microsoft Exchange (Silva, 2012).

3.3 Assessing and Evolving BYOD

Based on the analysis and the framework outlined in the previous sections, a multidimensional approach for assessing the critical phases of BYOD policy implementation can be suggested:

- Risk Analysis to validate the strength of the BYOD policy; the following four key factors should be considered:
  - Social Experience – customers' and employees' satisfaction
  - Operational Efficiency – business process continuity and evolution
  - Financial Viability – forming business metrics over the lifecycle of the BYOD program
  - Technical Practicality – risk avoidances and risk management: how to prevent mobile security threats, how to handle disastrous events.
- Legal Issues and Privacy Concerns – several possible scenarios can be adopted to allow companies to benefit from being BYOD friendly while properly balancing the company's data security and compliance needs with employee preferences and concerns. The most common choices can include:
  - Mobile user utilizes personal smartphone for personal and work purposes. The company uses written contracts and MDM solutions that offer employees device flexibility and optimal network access in exchange for giving up some control over their personal devices. The company owns the right to wipe out the corporate documents from the user's device in case it is lost or hacked. The personal information should be guaranteed not to be erased or modified by the company's reaction.
  - The company owns the devices and enables employees to use them for both work and personal purposes – this trend is known as COPE (company-owned, personally-enabled). All information on that device can be erased or modified any time according to corporate rules and regulations.
  - Limited network access and data storage abilities in order to improve data security and employee privacy. This approach eliminates the need to utilize MDM; instead MAM or MCM would be recommended options to enable secure access to corporate data (Finneran and Brashear, 2014).

For any of the above cases, the BYOD program should apply concurring containerization, separate interfaces for corporate and personal data, special settings for data syncing and backup, and supplementing device-level encryption for enhancing corporate data security.

4 CONCLUSIONS

The current generation of mobile users demand a high quality wireless experience all of the time. Mobile workers depend on it and when it is not delivered as expected, productivity drops, ultimately costing company efficiency, profits, and brand reputation. Therefore, the paper discusses the needs, the decision framework, and the quality of user experience to be at the heart of any BYOD strategy. The adopted corporate framework has to provide a consistent, predictable, frequently updated, and secure experience for all users of the utilized mobile platforms, devices, and/or applications.

Further work is planned in two directions: how consumerization of IT is aligning to the current business strategies and operation models, and how carriers' 3G and 4G infrastructures and vendor's specific mobile platforms operationally and functionally impact the consumerization of IT at the enterprises.

REFERENCES

AirWatch, 2013. *Mobile Content Management: Top 10 Considerations*, Atlanta, GA, USA. Retrieved on 4/1/2014 from: http://www.air-watch.com/downloads/resources/white-paper-mobile-content-management.pdf

Appcelerator, Inc., 2013 *State of the Mobile Enterprise – Q1 2013 Mobile Enterprise Report,* Retrieved on 02/1/2014 from: https://www.appcelerator.com/enterprise/resource-center/research/q1-2013-mobile-enterprise-report/

ComputerworldUK, 2012. *BYOD Makes Employees Work Extra 20 Hours Unpaid*. Retrieved on July 1, 2013 from: http://www.computerworlduk.com/news/mobile-wireless/3377143/byod-makes-employees-work-extra-20-hours-unpaid/

Finneran, M., Brashear, J., 2014. *A Legal Perspective of BYOD: Building Awareness to Enable BYOD and Mitigate its Risk*, ZixCorp. Retrieved on March 25, 2014 from: http://go.zixcorp.com/rs/zixcorp/images/Zix%20Ebook_A%20Legal%20Perspective%20of%20BYOD.pdf

Forrester Consulting, 2013. *Exploring Business and IT Friction: Myths and Realities*, Cambridge, MA, USA

International Data Corporation IDC, 2012. *Worldwide Mobile Worker Population 2009-2013 Forecast and Worldwide Mobile Enterprise Management Software 2012-2016 Forecast and Analysis and 2011 Vendor Shares*, Retrieved on June 28, 2013 from: http://www.gotomypc.com/remote_access/images/pdf/How_to_Equip_Your_Company_for_the_New_Mobile_Workforce.pdf

iPass Inc., 2013. *The Enterprise Mobility Guide for IT Management and CIOs*, The iPass/MobileIron Mobile Enterprise Report, Redwood Shores, CA, USA

MobileIron, 2014. *Mobile Application Management*. Retrieved on 3/29/2014 from: http://www.mobileiron.com/en/solutions/mobile-application-management

Rowsell-Jones, Andy, Jones, Nick. 2012 *Checklist for Determining Enterprise Readiness to Support Employee-Owned Devices,* Gartner, June 18, 2012, ID: G000234127

Silva, Peter, 2012. *BYOD 2.0: Moving Beyond MDM,* F5 White Paper. Retrieved on 09/20/2013 from: http:// www.f5.com/pdf/white-papers/big-ip-apm-mobile-application-manager-white-paper.pdf

SPECIAL SESSION ON e-HEALTH SERVICES AND TECHNOLOGIES (EHST)

Responsive Universal Design with Universal User Profiles and CSS User Queries

Hao-Wen Yang and Hsing Mei
Department of Computer Science and Information Engineering, Fu-Jen Catholic University, Taipei, Taiwan
bardichezanber@weco.net, mei@csie.fju.edu.tw

Keywords: Responsive Design, Responsive Universal Design, Cascading Style Sheet, CSS User Queries, Universal User Profile.

Abstract: Accompanying the rapid progressing of information and communication technology, the number of web-based mobile applications grows significantly. The web page design does bring different user experiences on various mobile devices. The Responsive Web Design (RWD) was the first trial to deal with this issue. RWD aimed at designing adaptive web interface for different mobile devices. However, RWD consider only the difference of devices, but not the users, especially the needs of elders. This paper proposes the Responsive Universal Design (RUD), an enhancement of RWD, to consider both the devices and users. A unique Universal User Profile (UUP) is maintained for each user on device. Based on UUP, CSS User Queries functions are used to adjust the user interface dynamically with the best experience. RUD can increase the flexibility of web design, and improve user experience. A JavaScript library is implemented for build/load UUP, to parse user queries, and to apply CSS styles. This paper reports a critical step to create a universal and user-oriented web environment.

1 INTRODUCTION

Accompany with the fast growth of network and mobile technologies, no matter when or where, we can find people using mobile phone or tablet devices for Internet access. However, most of the web sites are designed for desktop. Mobile web sites are costly to design. In 2010, Ethan Marcotte coined the term Responsive Web Design (RWD) in his article. Through the adoption of CSS Media Queries, the RWD based web page can response different views for different devices. However, RWD could meet the needs for each individual user. In the meantime, the aging society is approaching rapidly, and the number of elders (senior citizens) in physical and mental degradation is fast increasing. Small text font and complicated workflow are difficult to learn and use for elders. Neither traditional web design nor RWD-design can solve the above problem. In other words, RWD based web sites are still not friendly for elder and children. The specialized web sites for elders do not provide personalized solution for user. In this paper, we extend the RWD to a meta-level as Responsive Universal Design, and designed two tools to solve the problems.

This research designed and implemented a Universal User Profiles (UUP) to save user's experiences, behaviour, and preference. Accompany with the proposed CSS User Queries, which is an extension from the CSS3 Media Queries, web designer and/or developer can transform an ordinary desktop based web pages, to an adaptive page, we named the design process Responsive Universal Design (RUD). The design principals of RUD are: Universal design, Personalize optimization, and User Centered Design. The goal is to construct an Internet environment that allows the elder, the young, and the physical and spiritual disabled people to browse the web intuitively. The second section of this paper introduces the background of this research and current web environment. In the third section, we present the main idea of RUD's design. The fourth section illustrates the CSS User Queries and Universal User Profiles (UUP). The following section is the experiment and result of this research. The final section is conclusion and future works.

2 BACKGROUNDS

This section introduces the two major background technologies of RUD: Responsive Web Design (RWD), and CSS3 Media Queries.

2.1 Responsive Web Design

The accessibility of a web page is depending on screen resolution and size of the device; it is different on a desktop, a notebook, a tablet, or a smartphone. In the past, there are two possible ways to deal with this situation. (1) Minimize the design layout for smallest size screen. (2) Develop various mobile version sites, and server detects mobile devices to redirect the sites to mobile version. From developer's view, the second solution is more popular. However, there are too many new device types available in recent years. To develop web site for each device is not only expensive but also difficult to synchronize and update for all versions.

Fortunately, Responsive Web Design (RWD), a flexible design paradigm, has been proposed for adapting multiple devices. RWD-based web sites can render a suitable layout based on different screen resolution or device type. RWD not only combined the front-end technologies, such HTML; CSS; and JavaScript, but also integrated the concepts of information flow and graphic design. The web page could detect the screen size to respond the content and layout dynamically. Hence, RWD provides the best visual effects for contents. It also eliminated the need for additional development and maintenance costs for each device. For users, RWD provides consistency content style and user interface. RWD has become one of the leading web design trends.

2.2 CSS3 Media Queries

CSS3 Media Queries is a relatively new feature of CSS. It is now the core technology that makes RWD possible. Through the Media Queries, developer can set HTML elements' size and style dynamically. RWD provides the design idea of Fluid Grid, which makes web page linearized on the phone, and coordinate with flexible images and texts. Figure 1 shows a single page made with CSS3 Media Queries on desktop and iPhone.

With various CSS styles, we can not only produce different page layouts but also make some change of page content. For example, the CSS attribute 'display' can choose either to control the content display, or not to control.

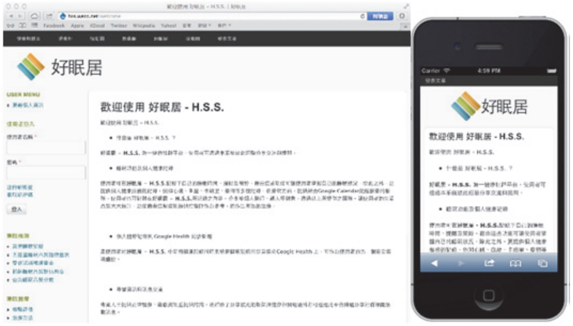


Figure 1: Example of Response Web Design with CSS3 Media Queries.

3 RESPONSIVE UNIVERSAL DESIGN

The user experience issues are receiving increasing attention. RWD supports adaptive design and style for various devices, but the difference of users is not considered. Here we propose a RWD-based extension which aiming at the user difference and experience. The Responsive Universal Design (RUD) is based on the following three principals:

- Universal Design

Similar to the concept of general universal design, we do not use RUD for specific groups. This design pattern does not only meet the need for elderly people, children and disabled persons, but also for the average person.

- Personalize Optimization

The main focus of responsive design is changing for difference. A web page is not suitable for anyone to browse. We need to do specific modification for different user, and provide them a most friendly experience.

- User Centered Design

Relative to "rendering on what devices", this design attaches more importance for "display for what type of user". The focus is returning to the user, and the design is initialized from the user's view.

Moreover, RUD is based as RWD and keeps all the advantages, e.g. lower development and maintain cost, high compatibility, etc.

4 UNIVERSAL USER PROFILES AND CSS USER QUERIES

To achieve the Responsive Universal Design (RUD), mechanisms to record user's experience and to query the preferred rendering are required. Here, we design the Universal User Profile (UPP) scheme and the CSS User Queries scheme.

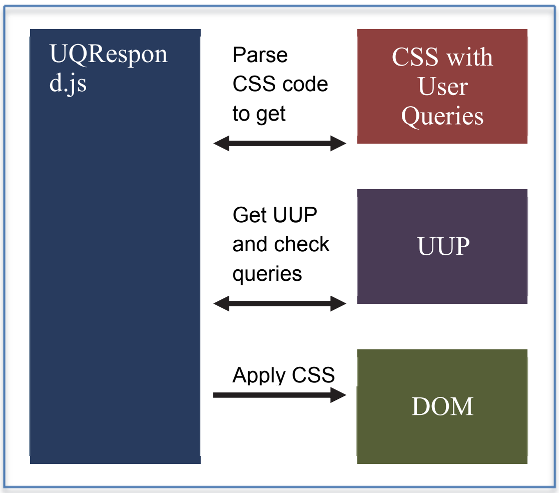
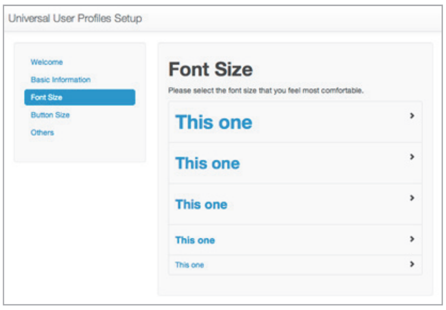


Figure 2: The relationship of JavaScript Library, User Queries, Universal User Profiles and Document Object Model (DOM).

4.1 Universal User Profile

RUD needs a mechanism to define the attributes and rules for the conditional query. All of user's attributes saved in a file dedicated to this task, called Universal User Profiles (UUP). Then, JavaScript can access and query for the user information to complete a RUD-based rendering. Figure 3 shows a setup screen layout of Universal User Profiles. Attributes in UUP include: Birthdate, Sex, Acceptable font size, Acceptable touch area, and preferred device.


Universal User Profiles Setup

Font Size

Please select the font size that you feel most comfortable.

Figure 3: Setup screen of Universal User Profiles.

- Birthdate

Birthdate is an important attribute in UUP and RUD mechanism. Due to the physical limitation, designers need pay more attention to elders and children when designing.

- Sex

This attribute is user's sex. We trust Internet is an open space and regardless of sex that anyone can browse, but the sex still will influence the user experience. We hope developer or designer can use this attribute for the style and composing, not for sexual content. It may also be useful for future data mining.

- Acceptable Font Size

UUP selected font size as a required attribute because most users are sensitive for the font size. This can ensure the experience of reading and make developers more convenient to set the font size.

- Acceptable Size of Touch Area

More and more people connect Internet through the touch-based devices like tablet or smartphone. The size of button directly affected the experience of the web site. Each person has his/her own definition of "easy-to-touch", for example, elderly people or children may need a bigger button for correct touch selection. Hence, UUP added the Acceptable size of touch area as attributes. The value of this attribute is a width of button, developer or designer can user this attribute with the Fitts' Law (in Human Computer Interaction) to deploy a fit button.

- Preferred Device

User's favoured devices and/or operating systems are also an important factor of user experience. UUP suggests give priority to the device that is using, to complete a consistency user experience. Also, if user/developers have any demands on special device, this attribute could be used to provide valuable information.

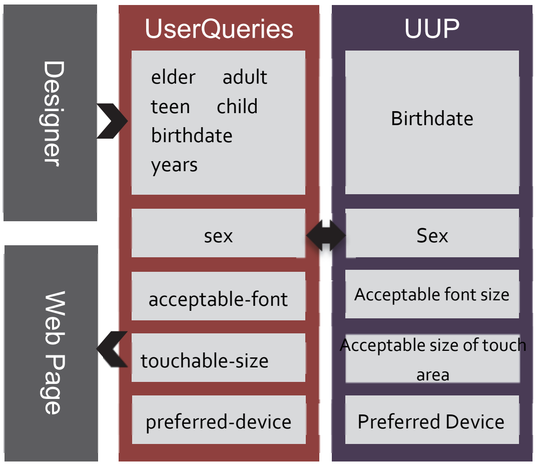
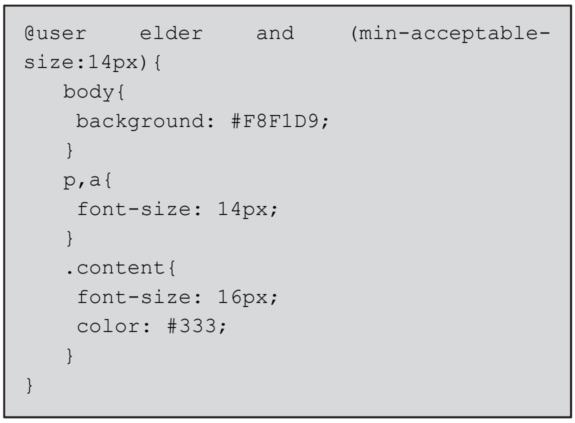


Figure 4: The model from the designer view.

4.2 CSS User Queries

Similar to the CSS3 Media Queries used in RWD, RUD also needs a mechanism for querying user status and applying the selected CSS code. Extending the format of CSS3 Media Queries, we developed the CSS User Queries. The usage of User Queries is again similar to Media Queries. Developer just tags @user in the CSS code and performs the query. It is simple and straightforward. Figure 5 illustrates a CSS User Queries example for an elder whose minimum acceptable font size is 14 pixels, and the background of web page will be light yellow, and the font size start from 14 pixels.



```
@user   elder   and   (min-acceptable-
size:14px){
   body{
    background: #F8F1D9;
   }
   p,a{
    font-size: 14px;
   }
   .content{
    font-size: 16px;
    color: #333;
   }
}
```

Figure 5: An example code of CSS User Query for an elder.

CSS User Queries provides 3 ways to query about user's age.

1. Elder, Adult, Teen or Child
Considering the distribution of users, users are divided to four age groups: elder, adult, teen and child. Developer can make a suitable adjustment for that age directly. The default area of each age are 0~14, 15~24, 25~64 and above 65 years old. Developers can adjust the border when they need.

2. Birthdate : yyyy (- yyyy)
If the developer likes to have a more precise query for age, 'birthdate' can be used as the parameter. Developer or designer can target a certain age to change the content or style of web sites. The parameter can be a range for some specify queries, e.g. '198x' or '1956 - 196x'.

3. Years : yy (- yy)
The third way is the current age representation. If developers want to query user's current age, they can user 'years_old' to complete this. It has similar usage as 'birthdate'.

Developer and designer can perform the following queries for the other UUP attributes.

- Acceptable font size: (acceptable-font : n px|em)
- Acceptable size of touch area: (touchable-size : n px|em)
- Preferred device: (preferred-device : device-type)
- Sex: Just add 'male' or 'female' in query.

4.3 JavaScript Library: Respond for User Queries (UQRespond.js)

To accomplish a RUD-based page, we designed and implemented a JavaScript library to build or load UUP, to parse user queries, and to apply CSS styles. This library is called UQRespond.js. It forked from an open source project named Respond.js, a JavaScript library to enable responsive web designs in browsers that don't support CSS3 Media Queries. Based on Respond.js, we edited it for User Queries and added core functions for UUP.

When user connected to a user query website first time, website can actively initiate or passively request user to setup the UUP file. UQRespond.js provide a simple and fast setup for that, just call the function "createUUP()", and it will start the UUP setup in the web page. With UUP setup, users do not need input parameters by themselves. In addition, UUP setup provides many examples for user to choose the best one. The Completed UUP will be saved as cookies or HTML5 web storages with the device. Hence, other web applications can access the same UUP.

We are also building a web service that user can be authenticated through OpenID. In this case, user can switch between difference devices, and maintain the user experience at same time. The relationship between UUP and user/devices is shown in Fig. 6.

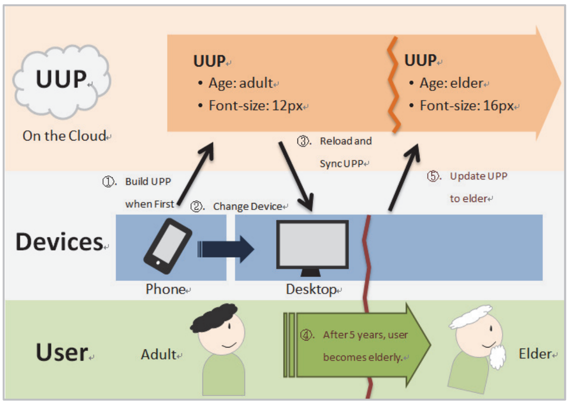


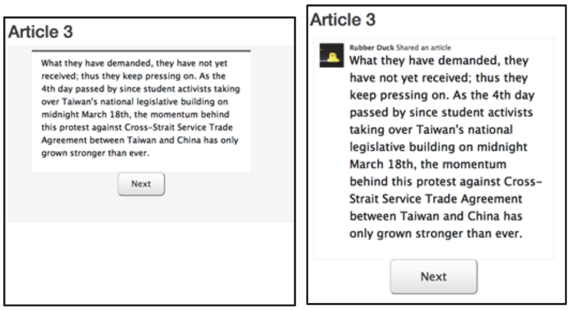
Figure 6: Schematic diagram of UUP, devices and user.

5 EXPERIMENT AND RESULT

This research proposed the Responsive User Design (RUD) mechanism for mobile web design. We designed and implement UUP Setup tool and library to verify the RUD. We also conduct real person experiment to investigate the user's behaviour with RUD. There are two objectives are expected from the experiment. First, User Queries can improve the user experience of reading and control a webpage. Among many User Experience parameters, we selected some quantifiable factors (e.g. reading time) to measure. Second, an UUP will bring a same fluent experience from web pages with different styles. We hope user can keep the improved user experience, which can be migrated from one RUD-based site to another.

There are two types of participants in this experiment, elders and teens. All participants have previous experience of using iPhone, iPad or Android devices. Before the experiment, subjects did not know the process of the experiment they will participate, and they will just receive an iPad with an opened web page. In the beginning, subject will enter their ages and set up a simplified UUP on the page. When the subject clicked the start button, the page shows an article and a next button. After read the article and clicked next, the web page will display a multiple-choice question about the article. Each experiment has four rounds of article and question.

The article in the experiment is a short and simple story or news that everyone can read easily. This research prepared two different CSS layouts to make articles look like reading from different web sites as shown in Figure 7, and randomly selected the styles with enable or disable User Queries in those four rounds. Our experimental pages save the touch events, reading time and status of each subject for analysing.



Article 3

What they have demanded, they have not yet received; thus they keep pressing on. As the 4th day passed by since student activists taking over Taiwan's national legislative building on midnight March 18th, the momentum behind this protest against Cross-Strait Service Trade Agreement between Taiwan and China has only grown stronger than ever.

Article 3

Rubber Duck Shared an article

What they have demanded, they have not yet received; thus they keep pressing on. As the 4th day passed by since student activists taking over Taiwan's national legislative building on midnight March 18th, the momentum behind this protest against Cross-Strait Service Trade Agreement between Taiwan and China has only grown stronger than ever.

Figure 7: Two different CSS layouts.

From the experiments, User Queries did improve participant's reading time. The User Queries enabled article reduced 12 percent reading time than disabled. It reduced 17 percent reading time for elder and 4 percent for teen. Totally 87.5 percent of participants improved reading efficiency with the UUP and User Queries.

Table 1: Experiment results.

		All	***Elders***	***Teen***
Selected font size (pixels)		26	27	26
Reading time (sec/article)	All	21.68	25.10	14.91
	User Queries enabled	20.27	22.73	14.60
	Default	23.10	27.48	15.24
Difference reading time in different styled pages (sec/article)	User Queries enabled	0.37	1.12	0.49
	Default	1.86	3.31	0.65
Number of touch events (times)		16	22	10

For the second objective, UUP will bring same experience from web pages with different style. The average reading time difference of two pages different with difference style is 1.86 seconds, and the average time difference become 0.37 seconds. It improved about 80 percent after enabled User Queries. As the result, UUP indeed improve the reading efficiency on the changing of the reading environment. This experiment is the first step of User Queries study, and we except User Queries will become an essential tool for designer to improve the user experience.

6 CONCLUSIONS

The idea of Responsive Universal Design (RUD) with CSS User Queries, and Universal User Profiles (UUP) was inspired and implemented from the Orange Technology. RUD can increase the flexibility of web design, and improve user experience. The JavaScript library is still under development and experiments are conducted on testbed continuously. We will release it as open source libraries to mobile web developer/designer community. We are also evaluating and planning the business model of the UUP web service. This paper reports the first step to create a universal and user-oriented web environment, and we trust this will be a trend for the web design in the future.

REFERENCES

Ethan Marcotte. (2010). Responsive Web Design. Retrieved from: http://www.alistapart.com/articles/responsive-web-design/

Hsing Mei, Vhien-Chang Hsu, and Wei-Ting Lai. (2008). Mashup a Reliable Virtual Health 2.0 Testbed Environment. Proceedings of the 4th Taiwan Conference on Software Engineering, 277-282.

Yan-Ting Lin, Chia-Hao Lin, Xiang-Lin Wen, Ching-Yang Hung and Hsing Mei. (2012). Joyful Health Space: An Age Friendly Testbed for Responsive Web Design. Orange Beneficence - International Design Conference of Smart Living for Elderly, 178-186.

F. Rivoal, H. Lie, T. Çelik, D. Glazman and A. van Kesteren. (2012). Media Queries, W3C Recommendation 19 June 2012. Retrieved form: http://www.w3.org/TR/css3-mediaqueries/

William Lidwell, Kritina Holden and Jill Butler. (2010). Universal Principles of Design, Revised and Updated: 125 Ways to Enhance Usability, Influence Perception, Increase Appeal, Make Better Design Decisions, and Teach through Design(2nd ed.). New York: Rockport Publishers.

Jesse James Garrett. (2010). The elements of user experience: user- centered design for the Web and beyond (2nd ed.). San Francisco: New Riders Press

Scott Jehl. (2011). Respond.js. Retrived from: https://github.com/scottjehl/Respond

The Effect of Touch Care for Baby by Mother

Yoko Hirohashi[1], Chieko Kato[2], Mayumi Oyama-Higa[3], Sang-jae Lee[4], Tomoe Sano[5] and Masato Ichikawa[6]

[1,2]Department of Nursing, NayoroCity University, W4-N8-1, Nayoro, Japan
[3]Chaos Technology Research Laboratory, Seta5-26-5, Otsu, Japan
[4]Department of Social Care, Nankai Vocational School, Chiyoda6-12-53, Takaishi, Japan
[5]Department of Education, Mukogawa Women's University, Ikebiraki-cho6-46, Nishinomiya, Japan
[6]Department of Nursing, Hokkaido University of Science, Teine-ku Maeda7, Sapporo, Japan
{hirohashi, chiekok}@nayoro.ac.jp, mhiga@chaotech.org, sjlee0729@yahoo.co.jp,
sano@mukogawa-u.ac.jp, ichikawa-m@hus.ac.jp

Keywords: Touch Care, Effect, Baby, Mother, Fingertip Pulse Wave.

Abstract: As the subject of child-rearing support has become more a part of society in modern times, in order to stimulate the spread of mother/child attachment formation through baby touch-care, the effectiveness of touch-care was verified by physical data collected by non-linear analysis, obtained from hospitalized mothers shortly after childbirth, as well as with mothers at home. Comparison of values of LLE and Autonomic Nerve Balance by means of fingertip pulse readings were taken before and after touch-care. As a result, the touch-care for baby with mother was effective to both baby and mother. The research suggests that: 1)the touch-care was effective to both baby and mother, 2) avoidance of using touch-care during hospitalization, 3) both mother and father acquiring skill in touch-care prior to childbirth, 4) after hospital discharge, implementation of touch-care in the home at play times and after bath times, are factors that contribute to emotional well-being.

1 INTRODUCTION

1.1 Issue of Support in Child-rearing

In the year 2005, along with the decline in total population, Japan recorded its lowest birthrate in the nation's history. If the birthrate decline continued at the same rate, by the year 2055 the total population would decrease to 90,000,000, the average number of children born would be 1.26 children per woman, the elderly population would be 40%, and the annual birthrate would be 500,000. Such a reduction in birthrate and the resulting decrease in overall population would create a far-reaching negative influence on Japan's economy due to a reduced labor force, impacting the pension system, medical services, healthcare, national insurance, etc. Such a threat to Japan's future existence is a cause for deep concern.

In the background of this rapid decline in birthrate, there is a great disparity between the desire of communities regarding a healthy rate of marriages, births/child-rearing, and the actual reality of the current situation. To eliminate this problem, it is important to create increased societal awareness from all viewpoints to effectively reconstruct solutions in institutions, government, etc., so that communities themselves can have hope in building a society where the citizens have peace of mind about marriage, bearing and raising children.

With the prevalence of the small family, a decrease in birthrate, a general breakdown of community, and a condition of parents feeling isolated and unable to properly raise children, parents need adequate knowledge, a positive attitude, and the support of an environment that assists them in the raising of children. However, this support must not come only from professionals in the infant-care and pre-school education sectors. In a broad sense, the care and support of parents and children is needed in all areas of society. For this reason, it is insufficient to delegate this task to "child-rearing support centers", but rather society as a whole must seek to provide support on a broad front. Therefore, the issue at hand is how to create an extensive network of individuals and locations throughout the country that will provide support for parents and children.

The goal is to aim for a society in which individuals in the community mutually benefit by sticking close in support of parents and their children. The desired goal is for society as a whole to do away with the solitary mindset where individuals think “I will raise my own child myself”, and to have citizens in each community mutually give support to parents, bringing about the trend of a child-rearing culture where “we will all work together to raise our children”.

1.2 Formation of a Parent/Child “Attachment” Relationship

During the time of a child’s infancy and preschool years, the development and acquisition of a “basic feeling of trust” is very important. This involves creation of an “attachment” between a young child and a specific adult in whom the child learns to trust and go to for answers. This trusting relationship becomes the basis through which the child gains a sense of ease and trust toward others throughout life.

According to John Bowlby (1907 - 1990), this “attachment” begins to form between a child and their caregiver (primarily the child’s mother) within the first 2 to 3 months after birth. In this relationship, the child begins to show a selective desire to be with the person caring for them, and the caregiver also experiences a greater awareness of affection for, and an intensified desire to nurture the child. Through the repetition of this mutual relationship, a bond of attachment is created and strengthened between the caregiver and the child. Because of this, in the development of the young child, for the sake of the child’s expression and vocalization of their own feelings, it is extremely important that the caregiver show an abundant measure of responsive affection through smiles, physical contact, etc.

1.3 Regarding Effectiveness of Touch-Care upon Newborns

During the early stage of a newborn’s life, it is said that a large number of necessary nerves in the brain develop through a remaining process (“Nerve Darwinism”). Therefore it is believed that through positive implementation of touch-care from the time shortly after birth, a heightened sensitivity and formation of a strong parent/child attachment is encouraged.

Research of touch-care for newborns was begun in the 1960’s, but these days is not limited to care of newborns only, but the practice of “Tactile Care ®” has also developed to involve people of every age, including the elderly, and is used as an effective means of aiding peace of mind as well as pain reduction. Also, touch-care is used to aid newborns (including premature newborns) in reduction of surplus levels of stress, and to stimulate proper child development. “Developmental Care” has been widely recognized as a means toward this goal, and through the specific means of “NIDCAP ® ” (Neonatal Individualized Developmental Care Program), the usefulness of touch-care has been demonstrated in promoting child development and formation of parent/child attachment.

1.4 Objective of Main Research

In sections I - III, the need in society for support of childcare, formation of parent/child attachment, and effectiveness of touch-care begun shortly after birth, were reported. In so doing, our research team verified the usefulness of childcare support in each locality, and the effectiveness of touch-care for infants as one means of attachment formation. From prior research regarding the fluctuation of fingertip pulse wave readings, in living creatures fingertip pulse wave readings showing “chaos” data from the brain were examined by means of non-linear analysis, and the effects of touch-care were investigated.

When a numerical value of mental activity can be derived by means of Largest Lyupanov Exponent (LLE) readings obtained through non-linear analysis of fluctuations in fingertip pulse wave data, verification is already obtained. It is our consideration that the practice of touch-care in some way or another affects the mental activity in mothers and infants, and that this can be known by fingertip pulse wave readings.

Also, from fingertip pulse wave fluctuation data, autonomic nerve balance (ANB) between sympathetic nerve priority vs. parasympathetic nerve priority can be understood. The autonomic nervous system is an accurate indicator of brain activity, and how autonomic nerve activity changes through touch-care can be obtained.

The aim of the main research is to verify, through this data, the effectiveness of touch-care for infants, and the means of application.

2 METHOD

2.1 Research Subject

The subjects for this research consisted of mothers hospitalized for childbirth, during the first 3-4 days after delivery, as well as mothers participating in childcare classes in various cities/towns, as well as mothers and young children during the childcare process in their locality. The target age for children in this study was under 1 year.

2.2 Research Period

December 2013 - February 2014

2.3 Research Location

“City A” General Hospital, childcare classes in “Town B” and, and the home of a participant in “City A”.

2.4 Research Method

First, explanation of the research was given to the participants, and a form with each participant’s consent for research was obtained. Next, pulse rates of the mother and child were measured for the first session. The mother’s pulse wave was monitored measuring the pulse at the tip of the second finger on the left hand using a finger cuff, and the pulse wave was measured for 3 minutes. The child’s pulse wave was obtained using a pediatric monitor attached to the child’s earlobe for 1 minute. Because of the challenge in getting a small child to sit still for long periods, measurement was set for a shorter time.

A birthing assistant (midwife) that helped with the research demonstrated the baby massage technique for the mother, using an infant size doll. Then the mother was asked to perform the same massage technique on her child. At the end of the massage period, the 2nd pulse wave reading was taken.

Also, a questionnaire for the participating mothers was utilized, which asked the mother two kinds of questions regarding 1) her impressions after having given the baby massage, and 2) how it affected her feeling of “attachment”. Questions regarding the mother’s impressions were as follows: “Did you experience any new realization of ‘touch-care’?” “Do you feel that you would like to continue this with your child? If so, what kind of ‘setting’ (location) would you prefer to use?” “What kind of feelings did you experience while doing the baby massage?” “Did you notice any place on your child where they seemed to enjoy the massage?” There were 26 questions regarding “attachment”, and response to these questions was divided into 4 categories: (nearly always, often, sometimes, and rarely).

2.5 Method of Analysis

For analysis of the mother’s responses, “Lyspect” measurement analysis computer software was used. For the analysis of the child’s responses, “BACS” measurement analysis software was used.

Data that was gathered comparing fluctuations in pulse wave rates before and after the experiment was examined by non-linear analysis using the computer software. The numerical values of the Largest Lyapunov Exponent (LLE) and Autonomic Nerve Balance (ANB) were examined for comparative study. It was judged that participants whose final readings, when compared to initial readings, showed a rise in LLE values indicated a rise in their mind’s sense of well-being. Also, it was judged that participants whose final readings showed a comparative decrease in ANB due to higher parasympathetic nerve activity indicated a mental reaction of relaxation.

Answers to questions regarding “attachment” were assigned a point value ranging from 4 to 1 for analysis purposes.

3 RESULT

3.1 Attributes of the Participants

In all, 25 mothers took part in the research. Among these, 15 were women tested during hospitalization, and the remaining 10 were women who were tested either at childcare classes or in their homes. The average age of the mothers was 28.8 ± 4.5 years. 7 mothers were tested 3 days after childbirth, 8 mothers were tested 4 days after childbirth, and 1 mother was tested 5 days after giving birth. 15 women were first-time mothers (primipara), and 10 women had previous births (multipara). All of the mothers had normal pregnancies lasting from 37 - 41 weeks, and none of the babies tested were underweight. Of the babies who were tested at home, ages were from 2 – 8 months old. In all, 10 babies were boys, and 15 were girls.

3.2 Regarding Results of Touch Care

Results in measurement of the mothers' fingertip pulse wave can be seen in the table 1.

After completion of the touch-care procedure, 16 of the 25 mothers showed a rise in Largest Lyapunov Exponent (LLE) and a drop in Autonomic Nerve Balance (hereafter referred to as ANB). In contrast, 9 mothers showed a drop in LLE and a rise in ANB. Within the model of increased LLE readings indicating a mental sense of well-being, and decreased ANB readings indicating a relaxed state, 64% of the mothers showed results suggesting that touch-care is effective.

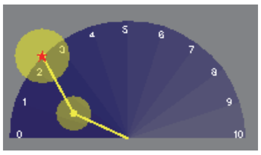


Figure 1: Case3's LLE.

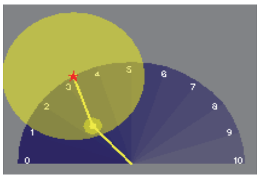


Figure 2: Case23's LLE.

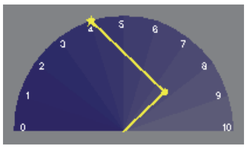


Figure 3: Case6's ANB.

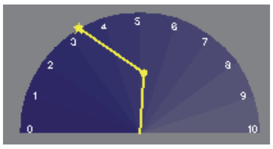


Figure 4: Case20's ANB.

Figure 1, 2, 3, 4: Cases where touch-care was effective.

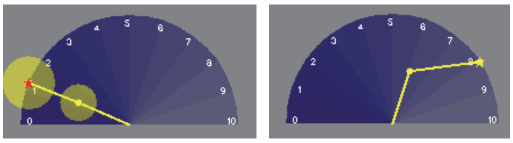


Figure 5: Case7's LLE and ANB.

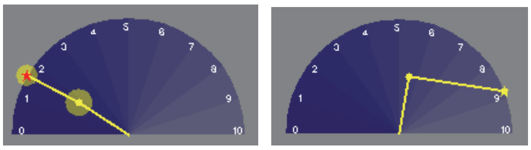


Figure 6: Case21's LLE and ANB.

Figure 5, 6: Cases effectiveness of touch-care could not be recognized.

Among mothers tested during hospitalization, 7 out of 15 women (46.7%) had results showing touch-care as "effective", and among mothers at home, 9 out of 10 women (90%) showed the same "effective" results.

Table 1: Data of all case (LLE and ANB).

	place	LLE ①	LLE ②		ANB①	ANB②		
1	HP	1.47	1.64	↑	4.4	5.6	↑	○
2	HP	1.35	2.47	↑	5.73	6.4	↑	○
3	home	2.13	2.62	↑	7.21	7.4	↑	○
4	home	2.4	4.69	↑	7.9	7	↑	○
5	home	2.37	2.2	↓	5.25	3.33	↓	○
6	HP	1.28	1.25	→	6.04	8.1	↑	×
7	HP	1.86	1.92	↑	3.78	4.67	↑	○
8	HP	2.05	1.09	↓	5.95	6.64	↑	×
9	HP	3.8	3.63	↓	7.43	7.83	↑	×
10	HP	1.55	1.77	↑	2.54	8.91	↑	○
11	HP	2.98	2.48	↓	4.34	6.3	↑	×
12	HP	2.44	2.31	↓	9.49	9.39	→	×
13	HP	0.91	0.63	↓	4.01	5.7	↑	×
14	HP	1.85	1.76	↓	6.3	5.59	↓	○
15	HP	2.37	3.17	↑	2.71	6.65	↑	○
16	HP	3.84	2.46	↓	6.96	7.68	↑	×
17	home	4.85	2.36	↓	3.6	6.46	↑	×
18	home	4.55	1.49	↓	3.6	1.97	↓	○
19	home	2.34	3.36	↑	5.79	6.41	↑	○
20	HP	2.46	1.88	↓	7.7	4.04	↓	○
21	HP	1.84	1.69	↓	5.6	8.8	↑	×
22	home	2.55	1.82	↓	5.57	4.81	↓	○
23	home	2.65	3.3	↑	5.35	7.4	↑	○
24	home	2.1	2.79	↑	5.35	7.04	↑	○
25	home	2.42	3.74	↑	9.49	8.82	↓	○

3.3 Regarding LLE and ANB Levels

In the 50 times that levels were measured, the average LLE level was a 2.38 reading. When looking at data from only mothers during hospitalization, the average LLE level was a 2.07 reading. Among mothers living at home, the average LLE level was a 2.84 reading.

In our previous research, 17 female students between the ages of 20 - 22 years old were tested, and when measurements were taken before and after food intake, the average LLE level was a 3.71 reading. Even when factoring in a decrease in levels immediately following food intake, these individuals had an LLE reading that was significantly higher than the average 2.07 LLE reading of mothers during hospitalization.

In all cases a trend was observed that immediately following the baby massage, ANB showed a gain in sympathetic nerve activity.

3.4 Attributes of Participants during Hospitalization Vs. at Home

Compared to participants during hospitalization, participants at home showed a significantly higher percentage of effectiveness regarding touch-care. (p = 0.045)

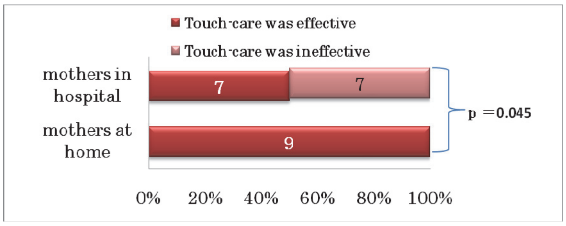


Figure 7: Comparisons between hospitalized mothers vs. mothers at home, regarding effectiveness of touch-care.

Compared to hospitalized participants, mothers at home showed a significantly higher percentage of awareness to where on the body their child enjoyed the baby massage. (p = 0.027).

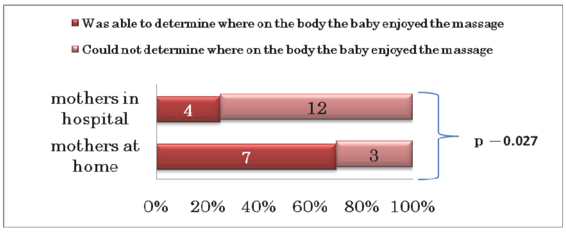


Figure 8: Comparisons between hospitalized mothers vs. mothers at home, regarding the response: "I could tell where on its body my baby enjoyed the massage".

Compared to hospitalized participants, mothers at home showed a much higher percentage of response that "I could understand the personality of my child", when answering the questionnaire. (p = 0.042).

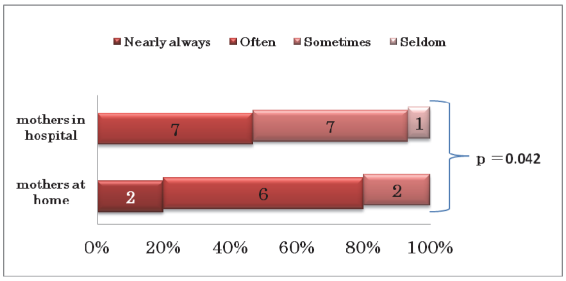


Figure 9: Comparisons between hospitalized mothers vs. mothers at home, regarding the response: "I understand the personality of my child".

3.5 Differences between First-time Mothers (Primipara) and Those with Prior Births (Multipara)

Primipara mothers, compared to multipara mothers, showed a much higher percentage of awareness that "I could tell where on its body my baby enjoyed the massage." (p = 0.014).

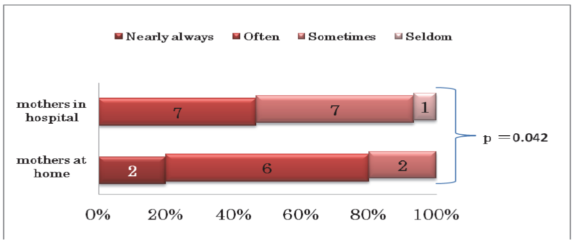


Figure 10: Differences between primipara mothers vs. multipara mothers regarding detection of where on the body the baby enjoyed massage.

Primipara mothers, compared to multipara mothers, showed a much higher percentage of response of "almost always", when asked how much time they would like to spend with just their baby. (p = 0.043).

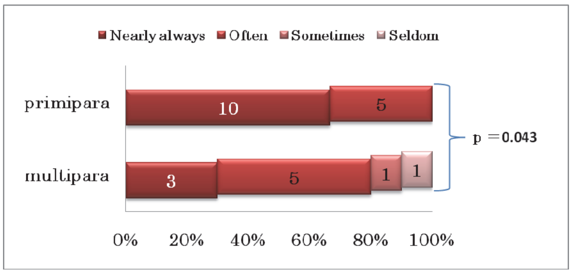


Figure 11: Comparisons between primipara mothers vs. multipara mothers regarding the response: "I want to spend time alone with my baby.

Primipara mothers, compared to multipara mothers, showed a much higher percentage of response saying, "I like to look into my child's eyes", when answering the questionnaire. (p = 0.027).

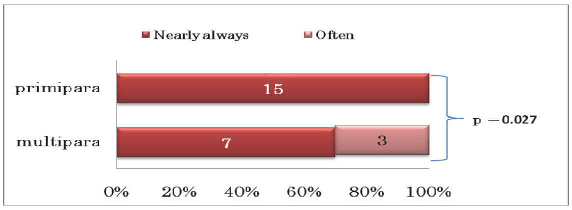


Figure 12: Comparisons between primipara mothers vs. multipara mothers regarding the response: "I enjoy looking into my baby's eyes."

3.6 Characteristics of the "Attachment" Measurement

The "Attachment" measurement had an average reading of 96.6 ± 5.8 points. We calculated all answer of each case to the points as “nearly always”=4 points, “often”=3 points, “sometimes”=2 points, “rarely”=1 point.

3.7 Impressions following Touch Care

Approximately 70% of the participants responded that they “mostly understood” regarding touch-care.

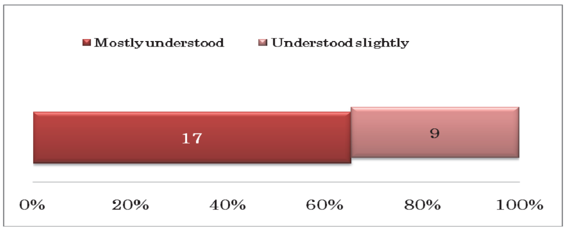


Figure 13: Level of comprehension with touch-care.

Approximately 70% of the participants responded that they “absolutely want to continue” touch-care.

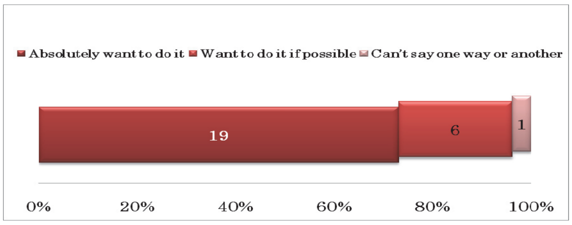


Figure 14: Desire to continue touch-care.

During the touch-care sessions, many mothers expressed feelings like “cute”, and when touching the baby made comments like “it’s a pleasant feeling”, “the baby is soft”, “the baby is warm”, etc. Negative comments such as “it’s scary”, “it makes me nervous”, “it’s a bother”, or “it’s tiring” were not expressed.

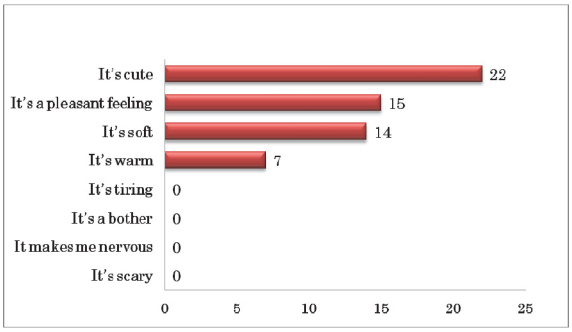


Figure 15: Impressions following touch-care.

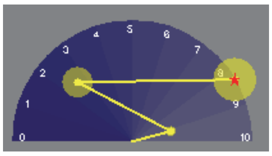
3.8 Regarding Results Taken from Baby Measurements

In 2 studies, baby fingertip pulse readings were taken before, immediately after, and 10 minutes after performing touch-care. Because nursing babies fingers are thin and because babies move around erratically, for pulse measurement a pediatric size cuff was attached to the baby’s earlobe. From previous research it was understood that the babies’ LLE readings were high. LLE readings just before and 10 minutes after touch-care were very high, but LLE levels were very low immediately following touch-care. In 2 studies, the obtained measurement results closely resembled this pattern both times. (Table2, Figure16, 17)

Because available time for set-up of equipment was limited, measurement of autonomic nerve balance (ANB) was not possible.

Table 2: Babies’ value of LLE.

Measurement time	**Baby A** **(boy, 5months)**	**Baby B** **(girl, 2months)**
before	**9.256**	**7.978**
immediately after	**2.625**	**2.433**
10 minutes after	**8.295**	**8.849**



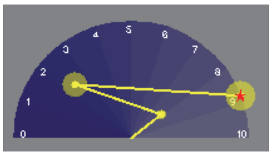


Figure 16: Baby A’s LLE. Figure 17: Baby B’s LLE.

4 CONSIDERATION

4.1 A Difference in Effectiveness of Touch Care during Hospitalization Vs. at Home

From measurements with this research, results showed that touch-care was effective in 64% of all the participant mothers. The heart of the mothers, who performed the massage, showed a lively attitude toward their children, and this was verified by fluctuations in the fingertip pulse wave measurements, as indicated by the LLE numerical values. However, a great discrepancy was noted among mothers tested 3 - 5 days after giving birth who showed an effectiveness of only 46.7% (less than half) vs. mothers at home, who showed an effectiveness of 90%.

When considering the cause of low effectiveness of touch-care among hospitalized mothers, post-partum depression that affects many women, also known as “baby blues”, could possibly be a factor. Due to drastic changes in hormone balance after giving birth, crying easily, irritability, anxiety about what childcare will involve, and other such symptoms are common. But even though the mother experiences the positive effect of giving baby massage and has pleasant feelings (“my baby is cute”) toward her child, the fact remains that the condition of her mind and body goes through a great deal of instability at this time. A mother usually recovers from this within 1 - 2 weeks after giving birth, but because our testing was done from the 3rd to 5th day after childbirth, it appeared that there were many mothers whose emotional state of mind was at a low point. The average level of LLE among these women was 2.07, an extremely low reading.

When considering these kinds of results, one other potential cause was the problem of the surroundings. The hospital rooms where mothers stayed had 6 women in a room, which made for a very tight space, and when baby bassinets were placed next to the mothers’ beds to allow the mothers and babies to be together in the same room, this made for an extremely cramped environment. In addition, due to the influence of an increasing societal trend toward more personal privacy, mothers completely closed the curtains around their beds, creating a lifestyle of isolation during hospitalization. Room lighting was rather dim, and the cramped space became a “boxed-in” environment. It is believed that these factors very possibly exerted an influence on the LLE readings of the mothers.

Also, it seems that the tendency of sympathetic nerve priority following the massage session was due to the mental concentration and nervousness of mothers who were practicing the massage for the first time. Since fatigue and hormonal changes after childbirth put an increased load upon a mother’s mind and body, it is thought that it would be better to avoid situations where the degree of nervousness is elevated.

With the passage of time after childbirth, the mother/child attachment increases, and a bond of trust is established. These factors likely explain the reason for the high level of effectiveness of touch-care with mothers at home. Another influence for consideration is that included in this group were mothers who had participated in childcare classes numerous times, and had already used baby massage repeatedly. Also, the mothers’ recognition of what part of the body their babies enjoyed being massaged, as well as comprehension of their babies’ personalities were linked to repeated usage of baby massage and observation of their babies over an extended period of time.

4.2 Differences between Primipara Vs. Multipara Mothers

Among primipara (first-time) mothers, numerous women responded by saying that they could determine what part of the body their child enjoyed being massaged, they enjoyed spending time alone with their child, and enjoyed looking into the eyes of their child. Results of this research showed that primipara mothers, compared to multipara mothers, took a longer amount of time in physical contact with their child, and had an environment where they focused a large amount of attention on their child.

4.3 Regarding Attachment

The emotional well-being of mothers during hospitalization was rather poor, nevertheless their response to questions regarding attachment averaged an extremely high reading of 96.6 points on a scale with a maximum 104 points possible. The following responses were especially prevalent: “I can sense the love of my child”, “I think my child is cute”, “I feel that I want to have a close relationship with my child”, “I like watching the gestures my child makes”, etc. Even though this research immediately followed childbirth, it was clear that there existed a strong level of mother/child attachment.

However, although a deep and healthy attachment of mothers toward their newborn babies was clearly visible, there were instances where the physical data taken from the fingertip pulse wave fluctuations did not correspond. Researchers grasped the fact that at times the actual participants themselves had a conscious awareness of the effects of touch-care, and at other times only had a mere outward observance without understanding parts of it. It was determined that by using fingertip pulse wave data objectively, aid in the care of emotional and physical health after childbirth is possible.

4.4 Impressions after Application of Touch Care

70% of the participants felt they understood touch-are well, and responded that they very much wanted to use it, experienced "cute"… "soft"… "feels good"… "warm" and other such pleasant feelings toward their child, and that they were conscious of the positive effect touch-care had on them.

When asked in what setting they would like to use touch-care, in addition to "play time", many women responded that they would use it after bath time. Recently, it is reported that in many families the father has assumed the role of caring for the baby's bath. Consequently, it is desirable that fathers also acquire ability to practice touch-care and form a bond with their child.

4.5 Effectiveness of Touch Care for Baby

It is known that shortly after birth there are large fluctuations in babies' fingertip pulse wave readings. It is said that this condition is a baby's adaptation to the outside world in the growth process. At 3 years of age LLE levels drop precisely, and because of this clear one-time drop the "3 year-old myth" is corroborated. However, it is considered that because the subjects of this research were children at the ages of 5 months and 2 months after birth, this period reflected very high LLE levels.

It is thought that the reason infant LLE levels drop sharply immediately following touch-care is due to the infant's "internal concentration" focus resulting from the touch-care session. This drop could be due to the condition of the infant's concentration regarding the sensations it experiences and the interaction with its mother, thus causing an "internal concentration" leading to the sharp drop in LLE levels.

10 minutes after touch-care, infant LLE levels returned to the initial high reading, but this type of fluctuation is thought to be a condition of the infant's growth development and enjoyment of the touch-care stimuli.

Although the infants spoke nothing, the physical data gave evidence that baby massage is effective. But because the subjects were limited to only 2 studies, it is needful for further testing to be done in order to more fully verify the findings.

It was determined that touch-care was effective for both mothers and their babies.

5 CONCLUSIONS

It was verified that baby touch-care is effective in aiding the mind/body health of mothers. Though examples are few, when making an objective observation from physical data taken from the babies, it was clear that baby touch-care is effective. Because of drastic hormonal changes after childbirth and the cramped environment with inadequate space in hospital settings, touch-care during hospitalization is best avoided. Rather, childcare classes prior to childbirth where both father and mother together practice touch-care, as well as learning touch-care in the home after childbirth when life has returned to normal, is to be desired. However, the most effective use of touch-care for newborns would be if both the mother and her husband could take their time and learn touch-care during the pregnancy, then after the birth, while preparing for discharge from the hospital, encourage a resumption of touch-care so that all aspects of touch-care can be implemented as the mother and child go home. The cooperation with both midwives and local health care nurses is important. In these days when the importance of child-rearing support is being spoken of in communities, it is the conclusion of our research that in order for mothers and their babies to experience a lifestyle where they enjoy health of mind and body in the area where they live, it is needful for baby touch-care to be more greatly popularized.

REFERENCES

Oyama-Higa, M., Miao, T., 2006. Discovery and application of new index of cognitive psychology. In *2006 IEEE Conference on Systems, Man, and Cybernetics*. Proceedings.

Oyama-Higa, M., Tsujino, J., Tanabiki, M., 2006. Does a mother's attachment to her child affect biological information provided by the child? -Chaos analysis of fingertip pulse waves of children-. In *2006 IEEE Conference on Systems, Man, and Cybernetics*. Proceedings.

Oyama-Higa, M., Miao, T., Tanaka, K., Cheng, H., 2007. Development of a self-check system for mental health using a pulse wave mouse. In *2007 IEEE Conference on Systems, Man, and Cybernetics*. Proceedings.

Hirohashi, Y., Oyama-Higa, M., 2008. The use of a non-linear analysis of pulse waves to measure the impact of music therapy and animal therapy on psychiatric care. In 2*008 IEEE Conference on Systems, Man, and Cybernetics*. Proceedings.

Okumura, Y., Matsuo, H., 2011. The effect of baby massage on stress reactions in mother and infant. In *Health care for mother*.Proceedings.

Nishida, H., 2012. *Book of Neonatalogy*, Volume 4, Igakushoin. Tokyo

Akagami, N., Kano, R., 2012. Effect of baby massage for child care by mother. In *Magazine of health care for mother in Ibaragi prefecture*. Editorial committee of health care for mother in Ibaragi prefecture.

Special Session on Intelligent Systems and Business Analysis

Uncertainty Modeling in the Process of SMEs Financial Mechanism Using Intuitionistic Fuzzy Estimations

George L. Shahpazov, Lyubka A. Doukovska and Vassia K. Atanassova
Institute of Information and Communication Technologies, Bulgarian Academy of Sciences,
Acad. G. Bonchev str., bl. 2, 1113 Sofia, Bulgaria
atlhemus@abv.bg, doukovska@iit.bas.bg, vassia.atanassova@gmail.com

Keywords: SMEs Financial Mechanism, Credit Risk, Creditworthiness, Intuitionistic Fuzzy Sets, i-Fuzzification.

Abstract: In the present paper, we discuss the mechanism of bank support of small and medium-sized enterprises (SMEs). Analysis is made of the effectiveness of the bank's internal financial structural unit and hierarchy, and it is shown how the concept of intuitionistic fuzzy sets can be applied to the process of evaluating creditworthiness of the SMEs applications for bank loans, from the bank's perspective. The presented approach aims to yield estimations of the effectiveness of the process, taking consideration of the aspects of uncertainty, which is an inherent part of the processes of evaluation of applications for bank support and evaluation of the process itself.

1 INTRODUCTION

Supporting emerging and present legal entities as making a form of investment, such as financing SME sector involves substantial risk in general and particularly in emerging markets like Bulgaria. A significant portion of this risks results from the lack of business ethics in the market and a legislation, which doesn't support in particular this kind of investments. Results published in paper (Shahpazov, Doukovska, 2012), shows that the timing for financial support in Bulgarian SMEs from the manufacturing sector is perfect. The actual result lays on deep analysis of the sector, which forecasted a faster growth in the sector than local GDP growth during a 5-8 year period spread.

Over the same period, the share of service sector output in GDP is expected to raise from 61.5% - 63.4%.

Local agriculture sector is experiencing a boost in the last few years, and falls under the program of rehabilitation and modernization of value creating industries, as the main focus is to overturn present trade situation where the country imports more goods than it exports. The overall aim is to utilize the EU accession and its supportive instruments, local Government programs assistance, and financial institution involvement into accelerating growth processes and SMEs further development.

The above mentioned facts allow us to look for new techniques for intelligent analysis of the process of SMEs financial mechanism.

In paper (Shahpazov, Doukovska, 2013), an application of the apparatus of generalized nets is proposed for modeling of the mechanism of financial support of the SMEs.

The present work traces the most important steps of the process of evaluation of a business project proposal, applying for bank financing. It is a continuation of our previous research (Shahpazov, Doukovska, 2013). The research model is offered how the concept of intuitionistic fuzziness can be applied to the process of evaluating creditworthiness of the SMEs.

The evaluation follows a predefined hierarchy of the levels of the bank's decision makers, and sophisticated policies and procedures.

For the needs of our discussion, we make a relatively simple model, which takes into account which levels of the bank hierarchy receive and process the business applications for bank loans, which levels make funding decisions, and in case of uncertainty, which upper levels of the hierarchy are these applications directed to, for taking a decision at the higher level. This model is schematically illustrated on Figure 1.

In this highly regulated process, for each level of the bank's decision making hierarchy, we are interested to estimate and interpret in terms of intuitionistic fuzzy sets the share of successfully approved applications, the share of rejected applications and the share of those applications, which for various reasons, may exhibit certain uncertainty (e.g. high risk / high return of investment) and thus get forwarded from lower to upper level of bank hierarchy, being a higher authority in the decision making process.

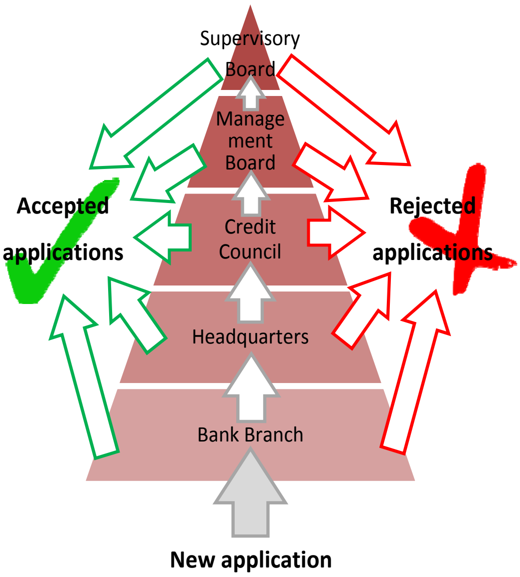


Figure 1: Diagram of the process of bank loan applictions review along the bank's decision making hierarchy.

2 SHORT REMARKS ON INTUITIONISTIC FUZZY SETS

Intuitionistic fuzzy sets (IFSs) were initially proposed by Atanassov in 1983 (Atanassov, 1983; Atanassov, 1986) as an extension of the concept of fuzzy sets, introduced by Zadeh in 1965 (Zadeh, 1965). The theory of IFSs has been extensively developed by the author in (Atanassov, 1991; Atanassov, 2012) and further developed by many other researchers worldwide.

In classical set theory, the membership of elements in a set is evaluated binary terms as either 'true' or 'false': an element either belongs or does not belong to the set. As an extension, fuzzy set theory permits the gradual assessment of the membership of elements in a set; this is described with the aid of a membership function valued in the real unit interval [0, 1].

The theory of intuitionistic fuzzy sets further extends both concepts by allowing the assessment of the elements by two functions, μ for the degree of membership and ν for the degree of non-membership, with which belong the element belongs to a set, where both these degrees and their sum are numbers in the [0, 1] - interval.

Speaking formally, if we have a fixed universe E and A is a subset of E, we can construct the intuitionistic fuzzy set A^*, so that:

$$A^* = \{\langle x, \mu_A(x), \nu_A(x)\rangle \mid x \in E\},$$

where $0 \leq \mu_A(x)$, $\nu_A(x)$, $\mu_A(x) + \nu_A(x) \leq 1$. In the case of strict inequality to the right, i.e.:

$$0 \leq \mu_A(x) + \nu_A(x) < 1,$$

there is a non-negative complement of the sum of membership and non-membership to 1, and this complement is denoted by $\pi_A(x) = 1 - \mu_A(x) - \nu_A(x)$ and usually called degree of uncertainty or hesitancy margin.

IFSs represent a true generalization of fuzzy sets, since in the partial case when the non-membership function fully complements the membership function to 1, not leaving room for any degree of uncertainty, is practically the case of fuzzy sets.

IFSs have different graphic representations, for instance linear, which bears resemblance with the graphic representation of fuzzy sets, radar-chart, or triangular, which reflects the specifics of the IFS. The standard linear graphic representation has the form of Figure 2, where both functions μ and ν are visualized as is.

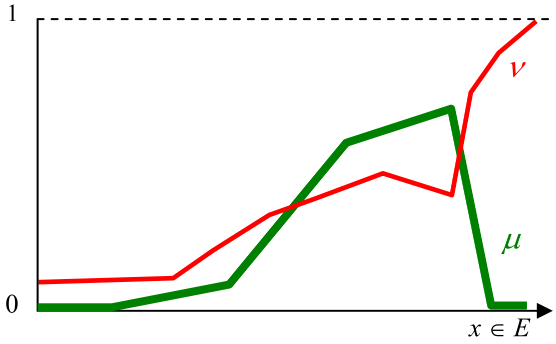


Figure 2: Standard graphical interpretation of IFSs.

However, together with the standard linear representation, a small modification of this graphics, as shown in Figure 3, was introduced (Atanassov, 1991) representing not the exact function ν, but the function $\nu^* = 1 - \nu$. It plots the non-membership function not in 'bottom-up' manner like the membership function μ, but in 'top-down' manner using its mirror image. Thus, we can very already well distinguish the formed in-between 'belt of uncertainty', which for every $x \in E$ complements the sum of $\mu_A(x)$ and $\nu_A(x)$ to 1. This modified linear representation of IFSs is probably the one most often used in practice.

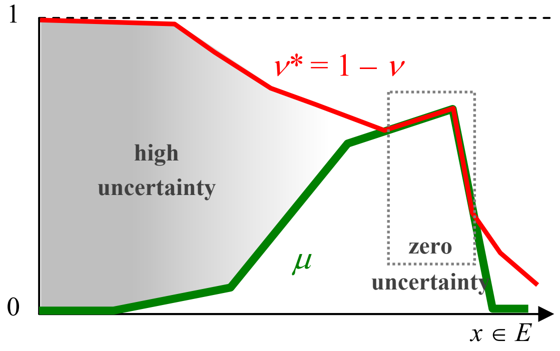


Figure 3: Modified graphical interpretation of IFSs.

3 MAIN RESULTS

As we mentioned above, the process of evaluation of every bank loan application passes through one or more (rarely more than three) levels of the bank's decision making hierarchy. Usually the decision about the approval or rejection of the applications is taken on the Branch level or the Headquarters level, however in certain cases when lower levels cannot take a categorical decision, the application is sent to the upper level.

Hence, it is of particular interest to trace the degrees of acceptance, rejection and uncertainty in taking the decisions on every bank hierarchy level, and for this purpose we can use a simple i-fuzzification procedure, analogous to the one given in (Atanassova, 2013), where from crisp data sets we can construct intuitionistic fuzzy data sets.

We can introduce intuitionistic fuzziness in these estimations, using two possible schemes, which are mathematically identical and can be used interchangeably, although visually they produce rather different results. In both cases, we will denote the levels of the bank's decision making hierarchy with the following denotations:

- Level 0 represents bank loan applicants,
- Level 1 is 'Branch' level,
- Level 2 is 'Headquarters' level,
- Level 3 is 'Credit Council' level,
- Level 4 is 'Management Board' level,
- Level 5 is 'Supervisory Board' level.

We will also agree to denote with μ_i, ν_i and π_i respectively, the number of applications, which on the i-th level are accepted, rejected or forwarded for decision to the level (i + 1), and with t – the total number of applications submitted for evaluation.

Obviously, in the top level of the Supervisory Board, $\pi_5 = 0$, as all applications that have reached this level must there get final resolution.

The whole process, interpreted in terms of IF estimations can be graphically illustrated in the following Figure 4.

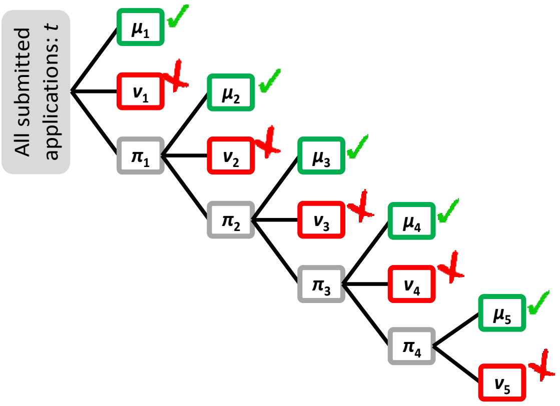


Figure 4: IF estimations of the performance of the different levels of decision making hierarchy during the bank loan applications review process.

First Scheme of i-Fuzzification. In the first scheme of i-fuzzification, on every level of the bank's decision making hierarchy, at a given moment of time, we estimate what percentage of the total number of submitted applications for evaluation have been approved, and, respectively, hitherto rejected. Let us denote these by M_i^1, N_i^1, $i = 1, \ldots, 5$, hence:

$$M_i = \frac{\sum_{k=1}^{i} \mu_k}{t}, \quad N_i = \frac{\sum_{k=1}^{i} \nu_k}{t}.$$

Second Scheme of i-Fuzzification. In the second scheme of i-fuzzification, on every level of the bank's decision making hierarchy, at a given moment of time, we estimate what percentage of the applications for evaluation, received from the lower level are approved, and, respectively, rejected, on that level. Let us denote these by M_i^2, N_i^2, $i = 1, \ldots,$ 5, hence:

$$M_i^2 = \frac{\mu_i}{\pi_{i-1}}, \quad N_i^2 = \frac{\nu_i}{\pi_{i-1}}.$$

Numerical Example. Graphical Interpretation of the Two Proposed i-Fuzzification Schemes. Let us give the following numerical example, which will make the differences between both proposed schemes easy to follow.

In given moment of time, let the following exemplary distribution of project applications along the levels in the bank's decision making hierarchy be observed, as shown on Figure 5.

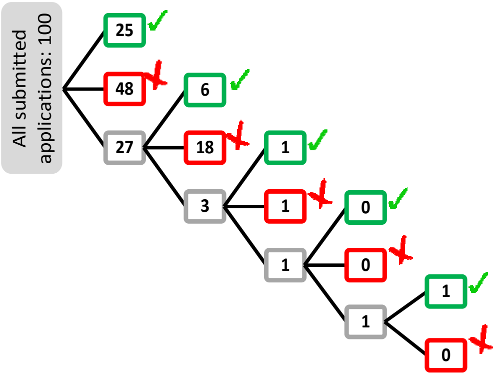


Figure 5: IF estimations for the numerical example.

Applying the first scheme of i-fuzzification over these data, will give the results in the following Table 1, as illustrated in Figure 6.

Table 1: Application of the first i-fuzzification scheme over the data from Figure 5.

	μ_i	ν_i	π_i
Level 1	25/100 = 0.25	48/100 = 0.48	27/100 = 0.27
Level 2	(25 + 6)/100 = 0.31	(48 + 18)/100 = 0.66	3/100 = 0.3
Level 3	(31 + 1)/100 = 0.32	(66 + 1)/100 = 0.67	1/100 = 0.01
Level 4	(32 + 0)/100 = 0.32	(67 + 0)/100 = 0.67	1/100 = 0.01
Level 5	(32 + 1)/100 = 0.33	(67 + 0)/100 = 0.67	0/100 = 0.00

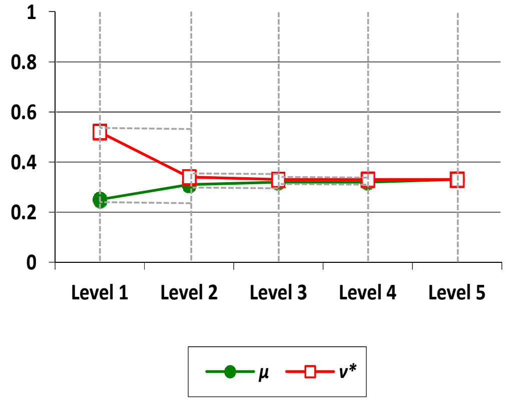


Figure 6: Interpretation of the first i-fuzzification scheme.

Applying the second scheme of i-fuzzification over these data, will give the results in the following Table 2, as illustrated in Figure 7.

Table 2: Application of the second i-fuzzification scheme over the data from Figure 5.

	μ_i	ν_i	π_i
Level 1	25/100 = 0.25	48/100 = 0.48	27/100 = 0.27
Level 2	6/27 = 0.22	18/27 = 0.67	3/27 = 0.11
Level 3	1/3 = 0.33	1/3 = 0.33	1/3 = 0.33
Level 4	0/1 = 0.00	0/1 = 0.00	0/1 = 0.00
Level 5	1/1 = 1.00	0/1 = 0.00	0/1 = 0.00

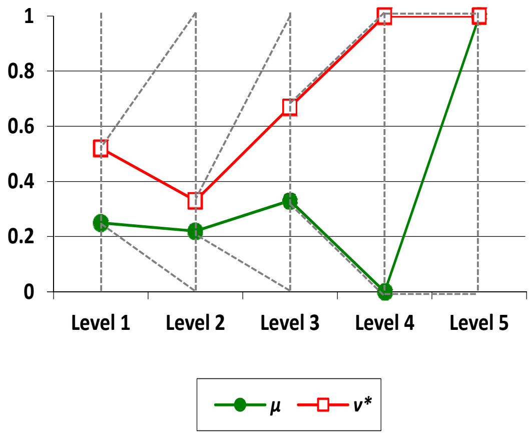


Figure 7: Interpretation of the second i-fuzzification scheme.

4 CONCLUSION

The comparison between both i-fuzzification schemes shows well that in the first scheme, at every level *i*, the [0, 1] - interval corresponds to the initial number of *t* submitted bank loan applications, and M_i^1, N_i^1, $i = 1, \ldots, 5$, are cumulative. In comparison, in the second scheme, on every upper level *i* we only operate with the IF evaluations for that level, and every time the degree of uncertainty from the lower $(i-1)^{\text{-th}}$ level is again re-normed to match the [0, 1] - interval (see the grey dotted lines).

Both approaches can be used interchangeably, and may prove useful in different situations, when it is necessary to evaluate the effectiveness of the different bank's internal financial structural unit as levels of the bank's decision making hierarchy.

ACKNOWLEDGEMENTS

The research work reported in the paper is partly supported by the project AComIn "Advanced Computing for Innovation", grant 316087, funded by the FP7 Capacity Programme (Research Potential of Convergence Regions) and partially supported by the European Social Fund and Republic of Bulgaria, Operational Programme "Development of Human Resources" 2007-2013, Grant № BG051PO001-3.3.06-0048.

REFERENCES

Atanassov K. (1983) Intuitionistic fuzzy sets, VII ITKR's Session, Sofia, June 1983 (in Bulgarian).

Atanassov K. (1986) Intuitionistic fuzzy sets. Fuzzy Sets and Systems. Vol. 20 (1), pp. 87-96.

Atanassov K. (1999) Intuitionistic Fuzzy Sets: Theory and Applications. Physica-Verlag, Heidelberg.

Atanassov K. (2012) On Intuitionistic Fuzzy Sets Theory. Springer, Berlin.

Atanassova, V. (2013) From fuzzy to intuitionistic fuzzy: Easy and lazy. Notes on Intuitionistic Fuzzy Sets, Vol. 19, 2013, № 3, 51-55.

Shahpazov G., L. Doukovska (2012) Structuring of Growth Funds with the Purpose of SME's Evolution under the JEREMIE Initiative, Proc. of the Second International Symposium on Business Modeling and Software Design - BMSD'12, Geneva, Switzerland, ISBN 978-989-8565-26-6, pp. 159-164.

Shahpazov G., L. Doukovska, K. Atanassov (2013) Generalized Net Model of the Methodology for Analysis of the Creditworthiness and Evaluation of Credit Risk in SMEs Financing, Proc. of the International Symposium on Business Modeling and Software Design – BMSD'13, Noordwijkerhout, The Netherlands, ISBN 978-989-8565-56-3, pp. 292-297.

Shahpazov, G., L. Doukovska (2013) Generalized net model of internal financial structural unit's functionality with intuitionistic fuzzy estimations. Notes on Intuitionistic Fuzzy Sets, Vol. 19(3), 111-117.

Zadeh L.A. (1965) Fuzzy Sets. Information and Control Vol. 8, pp. 333-353.

Significance of the Predictive Maintenance Strategies for SMEs

Mincho B. Hadjiski, Lyubka A. Doukovska, Stefan L. Kojnov,
Vladimir V. Monov and Vassil G. Nikov
*Institute of Information and Communication Technologies - Bulgarian Academy of Sciences,
Acad. G. Bonchev str., bl. 2, 1113 Sofia, Bulgaria*
hadjiski@uctm.edu, {doukovska, vmonov}@iit.bas.bg, slk@iinf.bas.bg, vasilnikov@abv.bg

Keywords: Predictive Maintenance, Advanced Integrated Maintenance Management System (AIMMS), Total Productive Maintenance (TPM), Reliability Centred Maintenance (RCM).

Abstract: The predictive maintenance is key to long-term profitability of a company in the manufacturing sector it can have a big impact on the supply, quality and price. The main methods are Total Productive Maintenance (TPM) and Reliability Centered Maintenance (RCM) and they must satisfy the requirements of the various small and medium-sized enterprises (SMEs). The paper identifies the barriers to the implementation of TPM within SMEs. Based upon our analysis a methodology for an integrated management system for predictive maintenance or the Advanced Integrated Maintenance Management System (AIMMS) is presented. The results presented in this paper show that AIMMS supports strategic decisions for predictive maintenance and it helps increase the equipment effectiveness by prioritizing the criticality of the equipment focusing on specific resources, increasing profits based on the Return On Investment (ROI).

1 INTRODUCTION

The predictive maintenance includes four stages: predictive diagnosis, estimation of potential looses, decision making for device maintenance and maintenance schedule arrangement. Technological diagnosis as the basis for predictive maintenance is established field of scientific and applied investigations. Predictive maintenance based on diagnosis, prolongs the life of machines and aggregates, reducing downtime, maintain optimal level of production, ensure compliance with the precise timing of delivery of production (raw materials, energy), allows for effective management of maintenance of facilities.

According to the International Standardization Organization (ISO) "Prognostics is time for estimation of damage and risk for one or several future damages", (ISO, 13381-I, 2004). Thus technological diagnosis can be understood as a process of estimation of Remaining Useful Life (RUL) before damage occurs, which is estimated based on the current status of the facility and last operating mode.

In world practice is increasingly accepted that predictive maintenance can play a key role in the long-term profitability of a company in the manufacturing sector with a major impact on timely delivery, product quality and its ultimate cost. The importance of maintenance increases in terms of increasing both the productivity and also the quality requirements which can only be achieved with a well-developed and organized maintenance strategy.

In this sense, according to recent advances in technology there have been developed and tested many methodologies, tools, techniques and strategies. The main methods are Total Productive Maintenance (TPM) and Reliability Centred Maintenance (RCM); the developed options are designed to meet the specific requirements of individual users, which are usually small and medium-sized enterprises (SMEs) operating in a very dynamic business environment. This paper focuses on identifying difficulties for the implementation of TPM within SMEs.

In recent years there has been increasing interest in the operation and management of industrial maintenance in a number of organizations. This is due to the increasing pressure on manufacturing organizations to meet customer and corporate requirements; the available equipment and productivity are central to achieving this goal. According to the authors' quote from a paper (Chan, 2005), "Recent trends show that on the whole many production systems do not function as expected when it comes to cost effectiveness in terms of their operation and maintenance. Lots of companies often operate with reduced capacities and reduced productivity, while the prices of their products are high.” A number of modern maintenance practices for technical support are designed to allow organizations to target strategic resources to achieve the maintenance tasks that are considered crucial for the effective and efficient operation of the equipment such as the Total Productive Maintenance (TPM). A number of organizations announce improvements in existing equipment, reliability and reduction in the maintenance costs after the implementation of TPM; the presented results are published in (Blanchard, 1997; Cooke, 2000). The benefits of TPM are often defined as an increase in the quality of products, availability of equipment and reduction of operative costs, according to the authors of (Cholasuke, 2004; Bohoris, 1995; Al-Najjar, 1996). In (Nakajima, 1988) it is assumed that TPM is used primarily because it integrates the functions of production and maintenance, but more importantly is that it redefines the role of operators and support engineers.

The author of (Nakajima, 1988) Nakajima is often accepted as the founder of the Total Productive Maintenance, (TPM); he assumes that the goal of TPM is to increase the efficiency of the equipment as well as to maximize the volume of production from this equipment. This is the result of an effort to achieve and maintain optimal conditions for the equipment in order to prevent unexpected failures, the speed decrease and the qualitative defects during the manufacturing process according to (Bamber, 1999). Via the application of TPM it is also expected to be possible to raise the moral of employees and their satisfaction from the job, suggesting the integration of workers into every aspect of the applied TPM. The majority of generally accepted definitions in TPM, used in (Barnes, 2002; Baglee, 2003; Baglee, 2010), are based on five main pillars outlined by Nakajima.

2 INTEGRATED MANAGEMENT SYSTEMS

Since 1996, the management of industrial sites is aware of the need of implementing systems to support all business processes (Dochain, 2008; The ANSVISA 95 Enterprise, 2005). These systems are known as Enterprise Resource Planning (ERP). Systems of this type are rapidly moving in large industrial complexes, but they remain almost completely cut off from existing DSC or SCADA. “Islands of automation” appear that are not at all related informationally and functionally to the general business management. Since the beginning of the 21st century this fragmentation begins to be overcome with the introduction of an intermediate layer of management – the Manufacturing Execution Systems (MES) (ISA the Instrumentation, Systems and Automation Society, 2005). These systems are a bridge between the technology management and the business management, so they perform a variety of roles for the operational management.

The purpose in (Staykov, 2013) is to make an analysis and assessment of the increasing need of using sophisticated software systems for managing business processes. The most important thing is the information and the success of every business depends on how fast and effective managers deal with information.

The structure of the integrated management systems primarily reflects the functions that must be implemented in a modern integrated management system of an industrial enterprise. It does not contain in itself a generalization of the methods and the tools for integration. This is essential, especially in the wide variety of engineering solutions that the world's leading providers offer. Therefore the accepted in 2005 standard ISA-95 is of great help to overcome internal fragmentation between control functions, built most often with different strategies, a specific vision and with a different magnitude. This is an international standard for the integration of business systems and control systems for the production processes shown in Figure 1.

This structure is *operative* and it includes next four hierarchical levels.

Level 1 and Level 2 are functionally grouped and comprise the management of individual devices and parameters. The generic name Process Control Systems (PCS) is accepted for them. The so formed block for Technological Management is treated as a generalized function in the specific problems (design, operation, adjustment) and it can be seen as composed of two levels. The timeline of the PCS-level are hours, minutes and seconds.

Level 3 comprises all tasks of operative management and it is accepted to be marked as MES. The main functions of this level are:

- Complete description of the production schedule;
- Management of production resources (people, equipment, materials);
- Specific dispatching of production with an

already formulated production schedule;

- Optimization of the production process;
- Monitoring of the overall production process;
- Analysis of the production process (quantity, quality, time schedule);
- Reliability and security of production;
- Reliable communication, acquisition and archiving the operational information;
- Implementation of the necessary operative instructions, as well as forming the tasks for 1-2 level (PCS).

The timeline of Level 3 are days, hours and minutes.

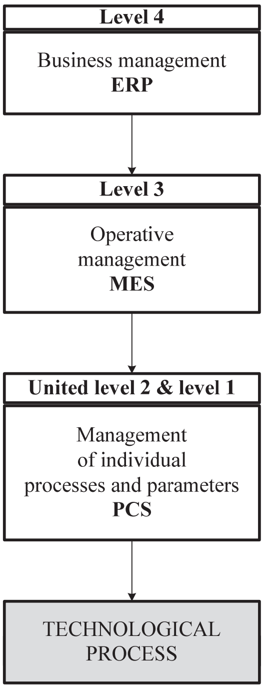


Figure 1: Production processes.

Level 4 is the level of business planning and logistics – ERP. The main functions of this level can be divided into two groups.

Functions directly related to the formation of the tasks for Level 3 (MES). These include:

- Production planning;
- Long-term production schedule;
- Define the requirements for manufactured products;
- Determine resource constraints, level of inventories, material consumption;
- Implement the supply chain of raw materials, production expedition, the overall logistics of the enterprise.

Functions relating to the overall business management, to cover:

- Financial accounting – books, fixed assets, payables, receivables, cash management;
- Management accounting – define cost and cost of production, cost control;
- Management of the supply chain – planning and schedule of supply, processing requests, purchases;
- Management of human resources – recruitment, training, remuneration, dismissal of staff;
- Project management – project-and-resources planning for their implementation, distribution of work, pricing and payment, timetables, implementation units, management activities;
- Management of customer relationships – sales and marketing, customer contact, market analysis;
- Comprehensive business analysis.

The timeline of the ERP-level are months, weeks or days. For purposes of this paper we are interested in the functionality of the MES-system.

The main merit of the standard ISA-95 is that it summarizes the experience of the leading academic institutions and companies worldwide in the way of overcoming the isolation between the technological management (PCS) and the business level (ERP) by creating a level of operative management MES. The ISA-95 standard defines the conditions for the integration of ERP and MES, on the one hand, and MES and PCS, on the other.

Standard ISA-95 decides primarily structural and information problems of integration. It to some extent does not repeal or does not replace the considerable experience in the application of various features in the specific automation on the four levels, accumulated for decades.

3 ANALYSIS OF THE BEST PRACTICES

1. In every industrial enterprise there are potential opportunities for improvement, which are in the range 1–10% by economy of energy resources and raw materials of 3–5%, 2–7% for increased production (http://www.automation.siemens.com, http://www.honeywellprocess.com/en-us).

2. Doing good business and in particular industrial management becomes more and more expensive.

3. Existing SCADA and DCS can not be changed frequently. The best approach is the old control systems to continue to work, by adding new technologies of control (Figure 2).

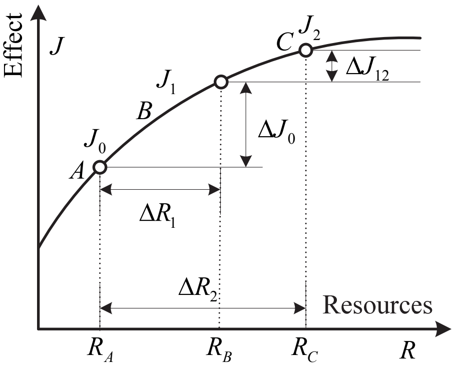


Figure 2: Process control effectiveness.

4. In deciding to develop the management system one must take into account the influence of many factors.

- The estimated economic impact of flexibility, quality improvement, increased productivity and reduced production costs;
- Spent money, effort and time;
- Reaction of the operating personnel;
- Security of humans and also of the facilities;
- Satisfy environmental and governmental regulations.

Unlike the position shown in Figure 2, we have a multicriteria problem.

5. Reengineering the system must take into account the specific circumstances:

- The degree of automation and information links;
- Different final goal to develop the system;
- The hardware and software from different vendors;
- Lack of prospects henceforth uses only one supplier.

6. Only a holistic approach can be effective. Information integration basically influences its realization.

7. The human factor is critical to the success of the integrated system. To do this:

- People need to understand the functions of the system;
- They must feel that they are part of the system with their expertise;
- They should not be overloaded by an extra loading;
- They must be sure that the system is safe and comfortable for them.

8. Outlines of some important drawbacks of private character:

- Monitoring of key industrial indicators – economic efficiency, safety and reliability is incomplete and it affects negatively the quality of decisions;
- Underestimated is the use of knowledge, especially on condition of the unanimous assessment that skilled and experienced-knowledge workers continuously retire without an equivalent replacement with new footage;
- Archival records are used less or not at all (Castiliio, 2013);
- The management of second and third level is more intuitive than based on online recommendations from DSS;
- Security systems (SSMs) are underestimated from terrorism and from unauthorized access, especially in terms of increasing wireless communications.

Figure 3 generalizes the management scheme, interpreting standard ISA-95 on functional level as multidimensional multistage system.

Provided, but not addressed in the standard division of the general part of technology management (PCS) at two levels (1 and 2) is considered an important structural feature. Each level is described as being of universal attributes, each of which is multi-dimensional and functionally different for each of the four levels. The following below attributes are considered.

- Hierarchical level;
- Tools for program-technical realization (SCADA, RLS, DCS, MES, ERP) *R*;
- Used mathematical models *M*;
- Criteria *J* and restrictions *L* in the optimization problem;
- Functions performed *F*;
- Control algorithms or decision making *A*;
- Information functions *I*;
- Operative and control actions *O*, *U*;
- Information disturbances *d*.

Each level receives its task from the preceding hierarchical level, including certain requirements in the form of specifications S_i, volumes V_i, criteria J_i, limitations L_i:

$$r_i = \left(S_i, V_i, J_i, L_i \right) \qquad (1)$$

The task is a vector comprising N_i components to each *i*-th level:

$$r_i = \left(r_{i1}, r_{i2}, \dots, r_{ij}, \dots, r_{iNi} \right) \qquad (2)$$

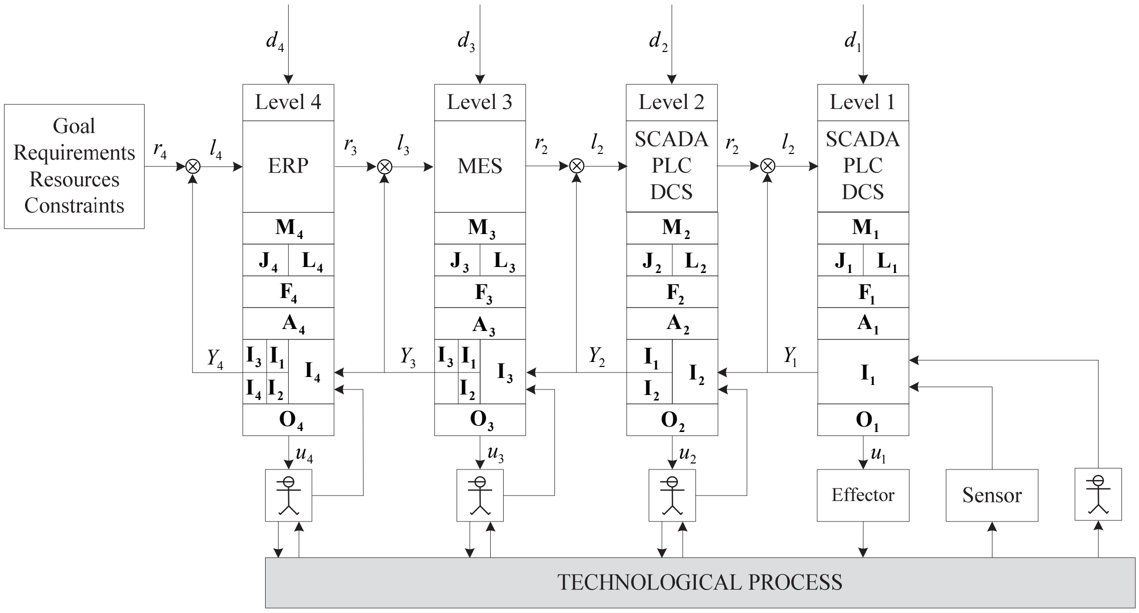


Figure 3: Generalized scheme.

The feedback y_{ij} is an assessment of the operation of each j-th element of i-th level E_{ij} and it has the same structure as the task r_i.

The aim of each level is to minimize the error e_i in terms of the specified criterion J_i with constraints S_i, V_i, L_i through controls u_i:

$$u_i = \arg \min_{S_i, V_i, L_i} J_i\left(e_i, u_i\right) \tag{3}$$

Criteria J_i and specifications S_i, volumes V_i and limitations L_i are vector and are different for each j-th E_{ij} element at level i:

$$\begin{aligned} J_i &= \left(J_{i1}, J_{i2}, \dots, J_{ij}, \dots, J_{iNi}\right) \\ S_i &= \left(S_{i1}, S_{i2}, \dots, S_{ij}, \dots, S_{iNi}\right) \\ V_i &= \left(V_{i1}, V_{i2}, \dots, V_{ij}, \dots, V_{iNi}\right) \\ L_i &= \left(L_{i1}, L_{i2}, \dots, L_{ij}, \dots, L_{iNi}\right) \end{aligned} \tag{4}$$

The controls u_i similarly are vectors, like the expressions (4), but each element E_{ij} may have several effects due to multidimensionality of the control:

$$u_{ij} = \left(u_{ij1}, u_{ij2}, \dots, u_{ijk}\right) \tag{5}$$

Operational impacts u_i can be both automatic controllers and also the actions of the operating personnel. A multistage system reflects both the hierarchical subordination of the lower level with respect to the upper one and also the inverse information influence of the lower level at the upper one. Each level has full access to the information at a lower level, but it adds also new information in the information flow upward as a result of the specific management and information processes at its own level.

4 CONCLUSIONS

Management of industrial complexes is gaining acceptance as a functional problem with multiple criteria the decision of which is possible only via a holistic approach.

Rapid development and application in DCS and SCADA receive a number of information technologies. Information integration is a key in building ERP, MES and PCS joint systems.

The methods of artificial intelligence (neural networks, fuzzy logic, genetic algorithms) and those based on knowledge (expert systems, autonomous agents, CBR) and especially the hybrid systems are still used only occasionally, but they possess a significant potential.

Generally ERP-MES-PCS systems are developed successfully to increase the competitiveness of industrial complexes in the globalized world.

The successful implementation of an approach to predictive maintenance taking into account the nature of activities in SMEs, it must be based on three main objectives. First, to allow the user to formulate a strategy of maintenance. Second, to convert the limited available data into knowledge to develop a strategy for maintenance. Third, to allow the user to record and measure the effect of the new strategy for support, to ensure that future decisions are based on facts and accurate data. The successful implementation of the formed in this way model would lead to the possibility the structure and the elements of the model to create efficient and effective strategy to support SMEs.

The main problem in developing a maintenance strategy is bounded by the lack of appropriate documentation and the ineffective analysis of available data. The reasons for this may be the lack of time for an understanding of the existing technology, and human relations in general. In particular, the staff must record their deeds and actions that relate to the problem; besides there must be measured the quantifiable, measurable benefits, which in the short term is difficult to achieve. Progress is possible if you use a simple system for recording and analysis that can be easily accessed and updated. The present research shows that technology is able to solve emerging needs but it requires innovative adaptations to solve the existing problems.

ACKNOWLEDGEMENTS

The research work reported in the paper is partly supported by the project AComIn "Advanced Computing for Innovation", grant 316087, funded by the FP7 Capacity Programme (Research Potential of Convergence Regions), partially supported by the European Social Fund and Republic of Bulgaria, Operational Programme "Development of Human Resources" 2007-2013, Grant № BG051PO001-3.3.06-0048 and partially supported under the Project № DVU-10-0267/10.

REFERENCES

http://www.automation.siemens.com

http://www.honeywellprocess.com/en-us

Al-Najjar B., 1996. Total Quality Maintenance. In *Journal of Quality in Maintenance Engineering* 2 (3), 4–20.

Baglee D., M. Knowles, 2010. Maintenance strategy development within SMEs: the development of an integrated approach, In *Journal Control and Cybernetics*, 39 (1), 275–303.

Baglee D., R. Trimble, J. MacIntyre, 2003. An investigation into the perceived barriers to total Productive Maintenance in small and medium enterprises. In *International Conference on Manufacturing Research*, University of Strathclyde, Professional Engineering Publishing.

Bamber C. J., J. M. Sharp., M. T. Hides, 1999. Factors affecting successful implementation of TPM: A case study perspective. In *Journal of Quality in Maintenance Engineering* 5 (3), 162–181.

Barnes D., 2002. The complexities of the manufacturing strategy formation process in practice. In *International Journal of Operations and Production Management* 22 (10), 1090–1111.

Blanchard B., 1997. An enhanced approach for implementing TPM in the manufacturing environment, In *Journal of Quality in Maintenance Engineering* 3 (2), 69-80.

Bohoris G. A., C. Vamvalis, W. Trace, K. Ignatiadou, 1995. TPM implementation in Land-Rover with the assistance of a CMMS. In *Journal of Quality in Maintenance Engineering*, 1 (4), 3–16.

Castillio O., P. Melinand, J. Kasprzyk, (Eds), 2013. *Recent Advantages on Hybrid Intelligent Systems*, Springer.

Chan F., F. Lau, S. Chan, S. Kong, 2005. Implementation of Total Productive Maintenance: A case study. In *International Journal of Production Economics* 95(1), 71–94.

Cholasuke C. R., R. Bhardwa, A. Jiju, 2004. The status of maintenance management in UK manufacturing organisations: results from a pilot survey. In *Journal of Quality in Maintenance Engineering* 10 (1), 5–15.

Cooke F. L., 2000. Implementing TPM in plant maintenance: Some organizational barriers. In *International Journal of Quality & Reliability Management* 17 (9), 1003–1016.

Dochain D., W. Marquardt, S. Won, O. Malik, M. Kinnaert, J. Lunze, 2008. Monitoring and Control of Process and Power Systems: Adapting to Environmental Challenges, Increasing Competitivity and Changing Customer and Consumer Demands, In *IFAC World Congress*, Seoul, Korea, 7160–7171.

ISA the Instrumentation, Systems and Automation Society, 2005. *The ANSVISA 95 Enterprise — Control System Integration*, Research Triangle Park, NC, USA.

ISO 13381-I, 2004. *Condition monitoring and diagnostics of machines-prognostics – Part: General guidelines.* Int. Standard, ISO.

Nakajima S., 1988. *Introduction to TPM: Total Productive Maintenance*. London, Productivity Press.

Staykov B., T. Atanasova, V. Monov, L. Doukovska, 2013. The ERP Systems in Modern Business and Corporate Management, In *Proceedings of the International Symposium on Business Modeling and Software Design – BMSD'13*, Noordwijkerhout, The Netherlands, 281–285.

Artificial Intelligence Neural Networks Applications in Forecasting Financial Markets and Stock Prices

Veselin L. Shahpazov, Lyubka A. Doukovska and Dimitar N. Karastoyanov
Institute of Information and Communication Technologies, Bulgarian Academy of Sciences,
Acad. G. Bonchev str., bl. 2, 1113 Sofia, Bulgaria
veselin_georgiev@abv.bg, doukovska@iit.bas.bg, dkarast@iinf.bas.bg

Keywords: Artificial Neural Network, Business, Predicting Stock Prices, Review, Historic Development.

Abstract: The interest in using artificial neural networks (ANN's) for forecasting has led to a tremendous surge in research activities over time. Artificial Neural Networks are flexible computing frameworks and universal approximators that can be applied to a wide range of time series forecasting problems with a high degree of accuracy. Forecasting problems arise in so many different disciplines and the literature on forecasting using ANN's is scattered in so many diverse fields that it is hard for a researcher to be aware of all the work done to date in the area. There is an extensive literature in financial applications of ANN's. Naturally forecasting stock price or financial markets has attracted considerable interest and it has been one of the biggest challenges. This paper reviews the history of the application of artificial neural networks for forecasting future stock prices. From the introduction of the back-propagation algorithm in 1980's for training an MLP neural network by Werbos, who used this technique to train a neural network and claimed that neural networks are better than regression methods and Box-Jenkins model in prediction problems through the application of such technics to financial markets forecasting by pioneers in the field like White, Kimoto and Kamijo to the more recent studies of stocks prices in not only the biggest capital markets but also in some emerging and illiquid markets, we will look at the progress made in the past more than twenty five years of research.

1 INTRODUCTION

It is nowadays a common notion that vast amounts of capital are traded through the stock markets all around the world. National economies are strongly linked and heavily influenced from the performance of their stock markets. Moreover, recently the markets have become a more accessible investment tool, not only for strategic investors but for common people as well. Consequently they are not only related to macroeconomic parameters, but they influence everyday life in a more direct way. Therefore, they constitute a mechanism which has important and direct social impacts.

The characteristic that all stock markets have in common is the uncertainty, which is related with their short and long-term future state. This feature is undesirable for the investor but it is also unavoidable whenever the Stock Market is selected as the investment tool. The best that one can do is to try to reduce this uncertainty. The Stock Market prediction task divides researchers and academics into two groups those who believe that we can devise mechanisms to predict the market and those who believe that the market is efficient and whenever new information comes up the market absorbs it by correcting itself, thus there is no space for prediction (EMH). Furthermore they believe that the Stock Market follows a Random Walk, which implies that the best prediction you can have about tomorrow's value is today's value.

In literature a number of different methods have been applied in order to predict Stock Market returns. These methods can be grouped in four major categories: 1) Technical Analysis Methods, 2) Fundamental Analysis Methods, 3) Traditional Time Series Forecasting and 4) Machine Learning Methods. Technical analysts, known as chartists, attempt to predict the market by tracing patterns that come from the study of charts which describe historic data of the market. Fundamental analysts study the intrinsic value of an stock and they invest on it if they estimate that its current value is lower that it's intrinsic value. In Traditional Time Series forecasting an attempt to create linear prediction models to trace patterns in historic data takes place. These linear models are divided in two categories: the univariate and the multivariate regression models, depending on whether they use one of more variables to approximate the Stock Market time series. Finally a number of methods have been developed under the common label Machine Learning these methods use a set of samples and try to trace patterns in it (linear or non-linear) in order to approximate the underlying function that generated the data. The aim is to draw conclusions from these samples in such way that when unseen data are presented to a model it is possible to infer the to-be explained variable from these data. These methods have been applied to market prediction; particularly for Neural Networks there is a rich literature related to the forecast of the market on daily basis.

2 AN OVERVIEW OF ARTIFICIAL NEURAL NETWORKS

The most commonly used forecasting network structure of ANN's is the multi-layer feed forward network brain particularly, are composed of a number of interconnected simple processing elements called neurons or nodes. Each node receives an input signal which is the total "information" from other nodes or external stimuli, processes it locally through an activation or transfer function and produces a transformed output signal to other nodes or external outputs. Although each individual neuron implements its function rather slowly and imperfectly, collectively a network can perform a surprising number of tasks quite efficiently. This information processing characteristic makes ANNs a powerful computational device and able to learn from examples and then to generalize to examples never before seen.

A number of different ANN models have been proposed. Perhaps the most influential models are the multi-layer perceptrons (MLP), Hopfield networks, and Kohonen's self-organizing networks. Other popular network structures are radial-basis functions networks, ridge polynomial networks, and wavelet networks.

An MLP is typically composed of several layers of nodes. The first or the lowest layer is an input layer where external information is received. The last or the highest layer is an output layer where the problem solution is obtained. The input layer and output layer are separated by one or more intermediate layers called the hidden layers. The nodes in adjacent layers are usually fully connected by acyclic arcs from a lower layer to a higher layer. Figure 1 gives an example of a fully connected MLP with one hidden layer.

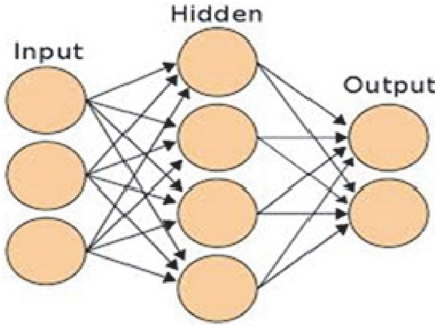


Figure 1: Example of a fully connected MLP.

For an explanatory or causal forecasting problem, the inputs to an ANN are usually the independent or predictor variables. The functional relationship, estimated by the ANN, can be written as:

$$y = f(x_1, x_2, \ldots, x_p) \quad (1)$$

where $x_1, x_2, \ldots, x_p$ are p independent variables and y is a dependent variable. In this sense, the neural network is functionally equivalent to a nonlinear regression model. On the other hand, for an extrapolative or time series forecasting problem, the inputs are typically the past observations of the data series and the output is the future value. The ANN performs the following function mapping:

$$y_{t+1} = f(y_t, y_{t-1}, \ldots, y_{t-p}) \quad (2)$$

where y_t is the observation at time t. Thus, the ANN is equivalent to the nonlinear autoregressive model for time series forecasting problems. It is also easy to incorporate both predictor variables and time-lagged observations into one ANN model, which amounts to the general transfer function model.

Before an ANN can be used to perform any desired task, it must be trained to do so. Basically training is the process of determining the arc weights which are the key elements of an ANN. The knowledge learned by a network is stored in the arcs and nodes in the form of arc weights and node biases. It is through the linking arcs that an ANN can carry out complex nonlinear mappings from its input nodes to its output nodes.

3 ANN APPLICATIONS AS FORECASTING TOOLS

Forecasting problems arise in so many different disciplines and the literature on forecasting using ANNs is scattered in so many diverse fields that it is hard for a researcher to be aware of all the work done to date in the area. One of the first successful applications of ANNs in forecasting is reported by Lapedes and Farber. Using two deterministic chaotic time series generated by the logistic map and the Glass Mackey equation, they designed the feed-forward neural networks that can accurately mimic and predict such dynamic nonlinear systems. Their results show that ANNs can be used for modelling and forecasting nonlinear time series with very high accuracy. After that a number of papers were devoted to using ANNs to analyse and predict deterministic chaotic time series with and/or without noise. Chaotic time series occur mostly in engineering and physical science since most physical phenomena are generated by nonlinear chaotic systems. As a result, many authors in the chaotic time series modelling and forecasting are from the field of physics.

There is an extensive literature in financial applications of ANNs. ANNs have been used for forecasting bankruptcy and business failure, foreign exchange rate, stock prices off course. Another major application of neural network forecasting is in electric load consumption study.

Many other forecasting problems have been solved by ANNs. A short list includes airborne pollen commodity prices, environmental temperature; helicopter component loads international airline passenger traffic macroeconomic indices, ozone level, personnel inventory, rainfall, river flow, student grade point averages, tool life, total industrial production, transportation, and wind pressure and wind pressure and many more.

4 ANN AND STOCK MARKET FORECASTING

The idea of using neural networks for predicting problems was first expressed in (Hu, 1964) which was used for weather forecasting. The absence of any learning method for multi-layer networks made it impossible to apply these networks to complex prediction problems. But in 1980's the back-propagation algorithm was introduced for training an MLP neural network. Werbos used this technique to train a neural network (Werbos, 1988) and claimed that neural networks are better than regression methods and Box-Jenkins model in prediction problems.

In recent years so many researches have been done on neural networks to predict the financial markets and the stock market changes in particular. Probably the first paper in the vast field, at least from today's perspective of stock market prediction with ANN's is (White, 1988), in which the main focus is to try to prove wrong the Efficient Market Hypothesis. In its simplest form, this hypothesis asserts that asset prices follow a random walk that is, the movement of an asset's price is completely unpredictable from publicly available information such as the price and volume history for the asset itself or that of any other asset.

One of the first efforts was by Kimoto and his colleagues in which they used neural networks to predict the index of Tokyo stock market (Kimoto *et al.*, 1990). They used several neural networks trained to learn the relationships between past values of various technical and economic indices for obtaining the expected returns of the TOPIX. The TOPIX is a weighted average of all stocks listed on the Tokyo Stock Exchange. The used technical and economic indices are: the vector curve (an indicator of market momentum), turnover, interest rate, foreign exchange rate and the value of the DJIA (Dow Jones Industrial Average). The desired output of the networks is a weighted sum, few weeks, of the logarithm of the ratio of the TOPIX at the end of week t to the TOPIX value at the end of week $(t - 1)$ and the desired output is a weighted sum of r_t for some weeks. The future extraction is not explained in this paper, except for the fact that some irregularity is removed and logarithm function is used before normalization.

In (Kamijo, Tanigawa, 1990) is proposed the use of "Elman recurrent net" (recurrent neural network is class of neural network, where connections between units form a directed cycle. This creates an internal state of the network which allows it to exhibit dynamic temporal behaviour) for predicting the future stock prices using extracted features from past daily high, low and closing stock prices. Unlike feed-forward neural networks, RNNs can use their internal memory to process arbitrary sequences of inputs. The method used tries to extract triangle patterns in stock prices which are seen on the daily high, low and closing graph. A triangle is usually seen as a beginning of a sudden stock price rise after that the high and low prices appear and the price oscillates for a period of time before the lines converge. The ANN is trained to recognize this pattern in the stock prices.

Matsuba uses a feed-forward NN with the last n stock index values as inputs and the next $N - n$ values as the outputs (Matsuba, 1991). This is an $N - n$ step ahead prediction. Thus, if the index for the $n^{\text{-th}}$ day is denoted by X_n then, the inputs are X_1, X_2, …, X_n and the outputs are X_{n+1}, X_{n+2}, …, X_N. If such a network is trained, any correlation between the index values for $n + 1$ through N^{th} day will be neglected. To ensure that this does not happen, the network is trained with errors between the desired and actual outputs in addition to the n inputs. These errors will then be $X_{n-1} - Y_{n+1}$, …, where Y is the output of the network. As the training proceeds this error will tend to zero and these additional inputs are not required in the testing phase.

In his work Freisleben used a simple feed-forward NN trained using past and present data to predict the value of the FAZ Index (Freisleben, 1992). Input data includes the moving average of past 5 and 10 weeks of the FAZ Index a first order difference of the FAZ Index and its moving average, the present bond market index and its first order difference and the Dollar-Mark exchange rate along with its first order difference. The value of the FAZ Index is predicted for the next week based on this data. The neural network is trained for the past M weeks and is then tested based on data for the next L weeks, where M is called the training window and L is called the testing window. For every successive prediction, the windows are shifted ahead and the network is retrained.

In (Azoff, 1994) is outlined that networks are computer program that can recognize patterns in data, learn from this and make forecasts of future patterns. At the time, there were just over 20 commercially available neural network programs designed for use on financial markets and there have been some notable reports of their successful application. However, like any other computer program, neural networks are only as good as the data they are given and the questions that are asked of them. Proper use of a neural network involves spending time understanding and cleaning the data: removing errors, pre-processing and post-processing. His book provides the knowledge that is required for the proper design and use of ANN's in financial markets forecasting – with an emphasis on futures trading.

In (Kaastra, Boyd, 1995) is provided a practical, non-technical introduction to designing a neural network forecasting model using economic time series data (16). The procedure of designing a model is divided into eight steps: 1) variable selection; 2) data collection; 3) data pre-processing; 4) training, testing and validation sets; 5) neural network paradigms; 6) evaluation criteria; 7) neural network training; and 8) implementation. Three major conclusions are made, the first being that researchers must have the time resources and patience to experiment mainly because of the nature financial markets. The second is that NN software must allow automated routines such as walk-forward testing, optimization of hidden neurons and testing of input variable combinations, either through direct programming or the use of batch or script files. And third the researcher must maintain a good set of records that list all parameters for each network tested since any parameter may turn out to cause a significant change in neural network performance.

In (Leung, Daouk, Chen, 2000) is conducted a research which focuses on estimating the level of return on stock market index. Given the notion that a prediction with little forecast error does not necessarily translate into capital gain, they evaluate the efficacy of several multivariate classification techniques relative to a group of level estimation approaches. Among the level estimation counter-parts, which forecast the level, are exponential smoothing are the multilayered feed-forward neural network and a probabilistic neural network.

With the introduction of electronic communication networks (ECN) as electronic trading systems facilitating trading of stocks and other financial products in the world's leading stock exchanges at first and later on other non-mainstream stock markets, and the constantly growing interest by both retail and institutional investors all around the world in stock's investing, the research in this field exploded. The advancement in computational and communicational power allowed researchers to develop models using artificial neural networks that are fed with real time data and capable to produce real time buy and sell signals.

In (Pan, Tilakaratne, Yearwood, 2005) is presented a computational approach for predicting the Australian stock market index – AORD using multi-layer feed-forward neural networks from the time series data of AORD and various interrelated markets. This effort aims to discover an effective neural network or a set of adaptive neural networks for this prediction purpose, which can exploit or model various dynamical swings and inter-market influences discovered from professional technical analysis and quantitative analysis.

Kalyvas attempts to predict the daily excess returns of FTSE 500 and S&P 500 indices over the respective Treasury Bill rate returns (Kalyvas, 2001). Then the author applies two different types of prediction models: Autoregressive (AR) and feed-forward Neural Networks (NN) to predict the excess returns time series using lagged values. For the NN models a Genetic Algorithm is constructed in order to choose the optimum topology. Data consists of 3275 daily observations of FTSE-100 index, UK T-Bill Rates and 3277 observations of S&P-500 index and US T-Bill Rates from 4 Jan 1988 until 12 Dec 2000. Finally he evaluates the prediction models on four different metrics and concludes that they do not manage to outperform significantly the prediction abilities of naive predictors.

In their study (Chen, Leung, Daouk, 2003) the authors attempt to model and predict the direction of market index of the Taiwan Stock Exchange, one of the fastest growing financial exchanges in the developing Asian countries (considered an emerging market). The probabilistic neural network (PNN) is used to forecast the direction of index return after it is trained by historical data. Statistical performance of the PNN forecasts are measured and compared with that of the generalized methods of moments (GMM) with Kalman filter. Moreover, the forecasts are applied to various index trading strategies, of which the performances are compared with those generated by the buy-and-hold strategy as well as the investment strategies guided by forecasts estimated by the random walk model and the parametric GMM models. They conclude that empirical results show that the PNN-based investment strategies obtain higher returns than other investment strategies examined in this study.

In (Kim, Lee, 2004) is compared a feature transformation method using genetic algorithm with two conventional methods for artificial neural networks. The genetic algorithm is incorporated to improve the learning and generalization abilities of ANN’s for stock market prediction. Daily predictions are conducted and their accuracy is measured. The authors use the proposed model to predict South Korea composite stock price index (KOSPI). The comparison of the results achieved by a feature transformation method using a genetic algorithm to other feature transformation methods shows that the proposed model performs better. Experimental results show that the proposed model reduces the dimensionality of the feature space and decreases irrelevant factors for stock market predictions.

In (Kim, 2006) is proposed a genetic algorithm approach to instance selection in artificial neural networks for financial data mining. He notes that artificial neural networks have preeminent learning ability, but often exhibit inconsistent and unpredictable performance for noisy data. In addition, it may not be possible to train ANN’s or the training task cannot be effectively carried out without data reduction when the amount of data is so large. The proposed model uses a genetic algorithm to optimize simultaneously the connection weights between layers and a selection task for relevant instances. The globally evolved weights mitigate the well-known limitations of gradient descent algorithm. In addition, genetically selected instances shorten the learning time and enhance prediction performance.

In (Madden, O’Connor, 2006) is evaluated the effectiveness of using external indicators, such as commodity prices and currency exchange rates, in predicting movements in the Dow Jones Industrial Average index. The performance of each technique is evaluated using different domain-specific metrics. A comprehensive evaluation procedure is described, involving the use of trading simulations to assess the practical value of predictive models, and comparison with simple benchmarks that respond to underlying market growth. In the experiments presented, basing trading decisions on a neural network trained on a range of external indicators resulted in a return on investment of 23.5% per annum, during a period when the DJIA index grew by 13.03% per annum.

In (Gosh, 2012) is presented a hybrid neural-evolutionary methodology to forecast time-series and prediction of the NASDAQ stock price in particular. The methodology is hybrid because an evolutionary computation-based optimization process is used to produce a complete design of a neural network. The produced neural network, as a model, is then used to forecast the time-series. The model identification process involves data manipulation and a highly experienced statistician to do the work. Compared to previous work, this paper approach is purely evolutionary, while others use mixed, mainly combined with back-propagation, which is known to get stuck in local optima. On the direction of model production, the evolutionary process automates the identification of input variables, allowing the user to avoid data pre-treatment and statistical analysis. The study proves the nimbleness of ANN as a predictive tool for Financial Time series Prediction. Furthermore, Conjugate Gradient Descent is proved to be an efficient Back-propagation algorithm that can be adopted to predict the average stock price of NASDAQ.

In (Chen, Du, 2013) are studied the interactions between social media and financial markets. The authors use a popular online Chinese stock forum

Guba.com.cn as well as traditional sentimental analysis methods, for each stock, they build a Social Behaviour Graph based on human's online behaviour, calculate key characteristics of the graph, and find out the correlations between trading volume/price and those characteristics. They make use of a back-propagation neural network to predict the trading volume and price of stocks from the Shanghai/Shenzhen Stock Exchange in China. Their method has achieved better outcome compared to the traditional trading volume/price based time series models. A trading strategy based on this method achieved 56.28% benefits for a period of three month, during which the stock index increased by only 1.17%.

5 CONCLUSION

This paper surveyed the application of neural networks to financial systems. It demonstrated how neural networks have been used to test the Efficient Market Hypothesis and how they outperform statistical and regression techniques in forecasting share prices. Although neural networks are not perfect in their prediction, they outperform all other methods and provide hope that one day we can more fully understand dynamic, chaotic systems such as the stock market.

Nowadays the technical advancement in computational power has served researchers to implement ANN's and obtain results faster and easier.

The Efficient Market Hypothesis is being heavily criticized and rejected mainly because of the fact that not all market participants possess the same amount of information and speed of access to the market and so on. This fact is encourages even more researchers to look for ways to predict the stock markets using the machine learning methods and artificial neural networks in particular.

Great deal of work has been done in the field since the late 1980's and progress has been substantial, putting ANN's in the centre of sophisticated models for predicting stock markets all around the world, from mainstream market indexes like the Daw Jones IA and S&P 500 through the Emerging markets of the BRIC to the less liquid financial markets in Eastern Europe, Latin and South America and the Middle and Far East.

ACKNOWLEDGEMENTS

The research work presented in this paper is partially supported by the FP7 grant AComIn №316087, funded by the European Commission in Capacity Programme in 2012-2016 and by the European Social Fund and Republic of Bulgaria, Operational Programme "Development of Human Resources" 2007-2013, Grant № BG051PO001-3.3.06-0048 from 04.10.2012.

REFERENCES

Azoff E. M., 1994. Neural Network Time Series Forecasting of Financial Markets. John Wiley&Sons.

Chen A. S., M. T. Leung, H. Daouk, 2003. Application of neural networks to an emerging financial market: Forecasting and trading the Taiwan Stock Index. Computers & Operations Research, Vol. 30, Issue 6, pp. 901–923.

Chen Z., X. Du, 2013. Study of Stock Prediction Based on Social Network, In: Proceedings of 2013 International Conference on Social Computing, September 8–14 Alexandria, VA, USA, pp. 913–916.

Freisleben B., 1992. Stock market prediction with back-propagation networks. In: Proceedings of 5th Intl. Conf. on the industrial and engineering application of artificial intelligence and expert systems, Germany, June 1992, pp. 451–460.

Gosh A., 2012. Comparative study of Financial Time Series Prediction by Artificial Neural Network with Gradient Descent Learning. Brain, Vol. 3 Issue 1, pp. 41–49.

Kaastra I., M. Boyd, 1995. Forecasting futures trading volume using neural networks Journal of Futures Markets, Vol. 15, Issue 8, pp. 953–970.

Kalyvas E., 2001. Using Neural Networks And Genetic Algorithms to Predict Stock Market Returns. Department of Computer Science, The University of Manchester, UK.

Kamijo K., T. Tanigawa, 1990. Stock market pattern recognition – A recurrent neural network approach Proc. IJCNN, San Diego, Vol. 1, pp. 215–221.

Kim K., 2006. Artificial neural networks with evolutionary instance selection for financial forecasting. Expert Systems with Applications, Vol. 30, Issue 3, pp. 519–526.

Kim W., B. Lee, 2004. Stock Market Prediction using artificial neural networks with optimal feature transformation. Neural Comput & Applic, Vol. 13, pp. 255–260.

Kimoto T., K. Asakawa, M. Yoda, M. Takeoka, 1990. Stock market prediction system with modular neural network. Proceedings of the International Joint Conference on Neural Networks, pp. 1–6.

Leung M. T., H. Daouk, A.-S. Chen, 2000. Forecasting stock indices: A comparison of classification and level estimation models. Int. J of Forecasting, Vol. 16, pp. 173–190.

Madden M. G., N. O'Connor, 2006. Neural Network Approach to Predicting Stock Exchange Movements using External Factors. In: Applications and Innovations in Intelligent Systems XIII, pp. 64–77.

Matsuba I., 1991. Neural sequential associator and its application to stock price prediction Proc. IECON 91, Japan, Vol. 2, pp. 1476–1479.

Pan H., I. Tilakaratne, J. Yearwood, 2005. Predicting Australian Stock Market Index Using Neural Networks Exploiting Dynamical Swings and Intermarket Influences, Journal of Research and Practice in Information Technology, Vol. 37, No. 1, pp. 43–55.

Werbos P. J., 1988. Generalization of back-propagation with application to a recurrent gas market model, Neural Networks, Vol. 1, pp. 339–356.

White H., 1988. Economic prediction using Neural Networks: The case of IBM daily stock returns. In: Proc. of the IEEE International Conference on Neural Networks, pp. 451–458.

Intercriteria Decision Making Approach to EU Member States Competitiveness Analysis

Vassia K. Atanassova[1], Lyubka A. Doukovska[1], Krassimir T. Atanassov[2,3] and Deyan G. Mavrov[3]
[1]Institute of Information and Communication Technologies, Bulgarian Academy of Sciences,
Acad. G. Bonchev str., bl. 2, 1113 Sofia, Bulgaria
[2]Institute of Biophysics and Biomedical Engineering, Bulgarian Academy of Sciences,
Acad. G. Bonchev str., bl. 105, 1113 Sofia, Bulgaria
[3]Prof. Dr. Asen Zlatarov University, 1 Prof. Yakimov Blvd., 8010 Burgas, Bulgaria
vassia.atanassova@gmail.com, doukovska@iit.bas.bg, krat@bas.bg, dg@mavrov.eu

Keywords: Global Competitiveness Index, Index Matrix, Intercriteria Decision Making, Intuitionistic Fuzzy Sets, Multicriteria Decision Making.

Abstract: In this paper, we present some interesting results derived from the application of our recently developed decision making approach to data from the World Economic Forum's Global Competitiveness Reports for the years 2008–2009 to 2013–2014. The discussed approach, called 'Intercriteria Decision Making', employs the apparatus of index matrices and intuitionistic fuzzy sets to produce from an existing multiobject multicriteria evaluation table a new table that contains estimations of the pairwise correlations among the set of evaluating criteria, called 'pillars of competitiveness'. Using the described approach over the data about WEF evaluations of the state of competitiveness of the 28 present EU Member States, certain dependences are discovered to connect the 12 'pillars', termed a 'positive' and a 'negative consonance'. The whole research and the conclusions derived are in line with WEF's address to state policy makers to identify and strengthen the transformative forces that will drive future economic growth.

1 INTRODUCTION

The present work contains a novel analysis of the most recent Global Competitiveness Reports (GCRs) of the World Economic Forum (WEF), produced from 2008–2009 to 2013–2014, aiming at the discovery of some hidden patterns and trends in the present Member States of the European Union. We use a recently developed method, based on intuitionistic fuzzy sets and index matrices, two mathematical formalisms proposed and significantly researched by Atanassov in a series of publications from 1980s to present day.

The developed method for multicriteria decision making (Atanassov et al., 2013) is specifically applicable to situations where some of the criteria come at a higher cost than others, for instance are harder, more expensive and/or more time consuming to measure or evaluate. Such criteria are generally considered unfavourable, hence if the method identifies certain level of correlation between such unfavourable criteria and others that are easier, cheaper or quicker to measure or evaluate these might be disregarded in the further decision making process. In particular, the approach has been so far applied to petrochemical industry, where the aim has been to reduce some of the most costly and time consuming checks of the probes of raw mineral oil, which have proven to correlate with other cheaper and quicker tests, thus reducing production costs and time needed for business decision making.

The present work is the first application of the developed approach in the field of economics. We have considered it appropriate to analyse our selection of data, in order to discover which of the twelve pillars (criteria) in the formation of the Global Competitiveness Index (GCI) tend to correlate. In comparison with related applications of the method, here, we do not conclude that any of the correlating criteria might be skipped, as in the petrochemical case study. We are interested however to discover dependences between the pillars, which could help policy makers, especially in the low performing EU Member States, to focus their efforts in fewer directions and reasonably expect on the basis of this analysis that improved country's performance against those pillars would positively affect the performance in the respective correlating pillars. Such correlation can be deemed reasonable to expect, as the twelve pillars are based on a multitude of indicators, some of which enter the GCI in two difference pillars each, as explained in the GCR's Appendix "Computation and structure of the Global Competitiveness Index" (and to avoid double counting, half-weight is being assigned to each instance).

This attempt to identify the correlations between the different pillars of competitiveness reflects WEF addressing the countries' policy makers with the advice to 'identify and strengthen the transformative forces that will drive future economic growth', as formulated in the Preface of the latest Global Competitiveness Report 2013–2014.

This paper is organized as follows. In Section 2 are briefly presented the two basic mathematical concepts that we use, namely, intuitionistic fuzzy sets and index matrices. On this basis, the proposed method is outlined. Section 3 contains our results from applying the method to analysis of a selection of data about the performance of the currently 28 Member States of the EU during the last six years against the twelve pillars of competitiveness. We report of the findings, produced by the algorithm and formulate our conclusions in the last Section 4.

2 BASIC CONCEPTS AND METHOD

The presented multicriteria decision making method is based on two fundamental concepts: intuitionistic fuzzy sets and index matrices. It bears the specific name 'intercriteria decision making'.

Intuitionistic fuzzy sets defined by Atanassov (Atanassov, 1983; Atanassov, 1986; Atanassov, 1999; Atanassov, 2012) represent an extension of the concept of fuzzy sets, as defined by Zadeh (Zadeh, 1965), exhibiting function $\mu_A(x)$ defining the membership of an element x to the set A, evaluated in the [0; 1]-interval. The difference between fuzzy sets and intuitionistic fuzzy sets (IFSs) is in the presence of a second function $\nu_A(x)$ defining the non-membership of the element x to the set A, where:

$$0 \le \mu_A(x) \le 1,$$

$$0 \le \nu_A(x) \le 1,$$

$$0 \le \mu_A(x) + \nu_A(x) \le 1.$$

The IFS itself is formally denoted by:

$$A = \{\langle x, \mu_A(x), \nu_A(x)\rangle \mid x \in E\}.$$

Comparison between elements of any two IFSs, say A and B, involves pairwise comparisons between their respective elements' degrees of membership and non-membership to both sets.

The second concept on which the proposed method relies is the concept of index matrix, a matrix which features two index sets. The theory behind the index matrices is described in (Atanassov, 1991). Here we will start with the index matrix M with index sets with m rows $\{C_1, \ldots, C_m\}$ and n columns $\{O_1, \ldots, O_n\}$:

$$M = \begin{array}{c|ccccccc}
 & O_1 & \cdots & O_k & \cdots & O_l & \cdots & O_n \\
\hline
C_1 & a_{C_1,O_1} & \cdots & a_{C_1,O_k} & \cdots & a_{C_1,O_l} & \cdots & a_{C_1,O_n} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\
C_i & a_{C_i,O_1} & \cdots & a_{C_i,O_k} & \cdots & a_{C_i,O_l} & \cdots & a_{C_i,O_n} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\
C_j & a_{C_j,O_1} & \cdots & a_{C_j,O_k} & \cdots & a_{C_j,O_l} & \cdots & a_{C_j,O_n} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\
C_m & a_{C_m,O_1} & \cdots & a_{C_m,O_j} & \cdots & a_{C_m,O_l} & \cdots & a_{C_m,O_n}
\end{array},$$

where for every p, q ($1 \le p \le m$, $1 \le q \le n$), C_p is a criterion (in our case, one of the twelve pillars), O_q in an evaluated object (in our case, one of the 28 EU Member states), $a_{C_pO_q}$ is the evaluation of the q-th object against the p-th criterion, and it is defined as a real number or another object that is comparable according to relation R with all the rest elements of the index matrix M, so that for each i, j, k it holds the relation $R(a_{C_kO_i}, a_{C_kO_j})$. The relation R has dual relation $\overline{R}$, which is true in the cases when relation R is false, and vice versa.

For the needs of our decision making method, pairwise comparisons between every two different criteria are made along all evaluated objects. During the comparison, it is maintained one counter of the number of times when the relation R holds, and another counter for the dual relation.

Let $S^{\mu}_{k,l}$ be the number of cases in which the relations $R(a_{C_kO_i}, a_{C_kO_j})$ and $R(a_{C_lO_i}, a_{C_lO_j})$ are simultaneously satisfied. Let also $S^{\nu}_{k,l}$ be the number of cases in which the relations $R(a_{C_kO_i}, a_{C_kO_j})$ and its dual $\overline{R}(a_{C_lO_i}, a_{C_lO_j})$ are simultaneously satisfied. As the total number of pairwise comparisons between the object is $n(n - 1)/2$, it is seen that there hold the inequalities:

$$0 \le S^{\mu}_{k,l} + S^{\nu}_{k,l} \le \frac{n(n-1)}{2}.$$

For every k, l, such that $1 \le k \le l \le m$, and for $n \ge 2$ two numbers are defined:

$$\mu_{C_k,C_l} = 2\frac{S^{\mu}_{k,l}}{n(n-1)},\ \nu_{C_k,C_l} = 2\frac{S^{\nu}_{k,l}}{n(n-1)}.$$

The pair constructed from these two numbers plays the role of the intuitionistic fuzzy evaluation of the relations that can be established between any two criteria C_k and C_l. In this way the index matrix M that relates evaluated objects with evaluating criteria can be transformed to another index matrix M^* that gives the relations among the criteria:

$$M^* = \begin{array}{c|ccc} & C_1 & \cdots & C_m \\ \hline C_1 & \langle \mu_{C_1,C_1}, \nu_{C_1,C_1} \rangle & \cdots & \langle \mu_{C_1,C_m}, \nu_{C_1,C_m} \rangle \\ \vdots & \vdots & \ddots & \vdots \\ C_m & \langle \mu_{C_m,C_1}, \nu_{C_1,C_m} \rangle & \cdots & \langle \mu_{C_m,C_m}, \nu_{C_m,C_m} \rangle \end{array}.$$

The final step of the algorithm is to determine the degrees of correlation between the criteria, depending on the user's choice of μ and ν. We call these correlations between the criteria: 'positive consonance', 'negative consonance' or 'dissonance'.

Let $\alpha, \beta \in [0; 1]$ be given, so that $\alpha + \beta \le 1$. We call that criteria C_k and C_l are in:

- (α, β)-positive consonance, if $\mu_{C_k,C_l} > \alpha$ and $\nu_{C_k,C_l} < \beta$;
- (α, β)-negative consonance, if $\mu_{C_k,C_l} < \beta$ and $\nu_{C_k,C_l} > \alpha$;
- (α, β)-dissonance, otherwise.

Obviously, the larger α and/or the smaller β, the less number of criteria may be simultaneously connected with the relation of (α, β)-positive consonance. For practical purposes, it carries the most information when either the positive or the negative consonance is as large as possible, while the cases of dissonance are less informative and can be skipped.

3 MAIN RESULTS

We ran the described algorithm over collected data from six WEF GCRs for the 28 (current) EU Member States. Here, we present only the results from the two extreme periods: years 2008–2009 and year 2013–2014, comparing them for $\mu_{C_iC_j}$ and $\nu_{C_iC_j}$ in Tables 1–2. Despite having the results with precision of 9 digits after the decimal point, we will use precision of 3 digits after the decimal point.

In Tables 1 and 2, all cells are coloured in the greyscale, with the highest values coloured in the darkest shade of grey, while the lowest ones are coloured in white. Of course, every criteria perfectly correlates with itself, so for any i the value $\mu_{C_iC_i}$ is always 1, and $\nu_{C_iC_i} = \pi_{C_iC_i} = 0$. Also, the matrices are obviously symmetrical according to the main diagonal. The twelve pillars are: 1. Institutions; 2. Infrastructure; 3. Macroeconomic stability; 4. Health and primary education; 5. Higher education and training; 6. Goods market efficiency; 7. Labour market efficiency; 8. Financial market sophistication; 9. Technological readiness; 10. Market size; 11. Business sophistication; 12. Innovation.

In the beginning, let us present in Table 3 some findings from the analysis of the six periods.

Table 1: Comparison of the calculated values of $\mu_{C_iC_j}$ for years 2008–2009 and 2013–2014.

μ	1	2	3	4	5	6	7	8	9	10	11	12
1	1.000	0.844	0.685	0.757	0.788	0.833	0.603	0.828	0.823	0.497	0.794	0.802
2	0.844	1.000	0.627	0.751	0.749	0.743	0.529	0.741	0.775	0.582	0.831	0.807
3	0.685	0.627	1.000	0.616	0.638	0.664	0.653	0.648	0.693	0.434	0.651	0.667
4	0.757	0.751	0.616	1.000	0.780	0.720	0.550	0.704	0.725	0.524	0.765	0.772
5	0.788	0.749	0.638	0.780	1.000	0.746	0.622	0.728	0.757	0.558	0.767	0.796
6	0.833	0.743	0.664	0.720	0.746	1.000	0.627	0.817	0.802	0.505	0.786	0.765
7	0.603	0.529	0.653	0.550	0.622	0.627	1.000	0.664	0.611	0.389	0.563	0.590
8	0.828	0.741	0.648	0.704	0.728	0.817	0.664	1.000	0.820	0.476	0.733	0.751
9	0.823	0.775	0.693	0.725	0.757	0.802	0.611	0.820	1.000	0.548	0.817	0.815
10	0.497	0.582	0.434	0.524	0.558	0.505	0.389	0.476	0.548	1.000	0.648	0.601
11	0.794	0.831	0.651	0.765	0.767	0.786	0.563	0.733	0.817	0.648	1.000	0.860
12	0.802	0.807	0.667	0.772	0.796	0.765	0.590	0.751	0.815	0.601	0.860	1.000

μ	1	2	3	4	5	6	7	8	9	10	11	12
1	1.000	0.735	0.577	0.720	0.807	0.836	0.733	0.749	0.854	0.503	0.804	0.844
2	0.735	1.000	0.479	0.661	0.749	0.677	0.537	0.590	0.786	0.661	0.804	0.799
3	0.577	0.479	1.000	0.421	0.519	0.558	0.627	0.675	0.550	0.413	0.548	0.556
4	0.720	0.661	0.421	1.000	0.730	0.683	0.590	0.563	0.677	0.497	0.712	0.690
5	0.807	0.749	0.519	0.730	1.000	0.735	0.622	0.632	0.775	0.579	0.815	0.847
6	0.836	0.677	0.558	0.683	0.735	1.000	0.749	0.712	0.788	0.466	0.759	0.751
7	0.733	0.537	0.627	0.590	0.622	0.749	1.000	0.741	0.685	0.399	0.624	0.624
8	0.749	0.590	0.675	0.563	0.632	0.712	0.741	1.000	0.712	0.497	0.688	0.680
9	0.854	0.786	0.550	0.677	0.775	0.788	0.685	0.712	1.000	0.526	0.810	0.831
10	0.503	0.661	0.413	0.497	0.579	0.466	0.399	0.497	0.526	1.000	0.611	0.598
11	0.804	0.804	0.548	0.712	0.815	0.759	0.624	0.688	0.810	0.611	1.000	0.873
12	0.844	0.799	0.556	0.690	0.847	0.751	0.624	0.680	0.831	0.598	0.873	1.000

Table 2: Comparison of the calculated values of $\nu_{C_iC_j}$ for years 2008–2009 and 2013–2014.

ν	1	2	3	4	5	6	7	8	9	10	11	12
1	0.000	0.114	0.241	0.140	0.140	0.077	0.275	0.116	0.116	0.458	0.148	0.127
2	0.114	0.000	0.304	0.156	0.190	0.167	0.365	0.220	0.180	0.384	0.127	0.138
3	0.241	0.304	0.000	0.265	0.265	0.209	0.204	0.270	0.225	0.495	0.270	0.241
4	0.140	0.156	0.265	0.000	0.108	0.140	0.294	0.201	0.169	0.381	0.138	0.111
5	0.140	0.190	0.265	0.108	0.000	0.135	0.233	0.198	0.164	0.378	0.156	0.130
6	0.077	0.167	0.209	0.140	0.135	0.000	0.209	0.090	0.095	0.397	0.114	0.127
7	0.275	0.365	0.204	0.294	0.233	0.209	0.000	0.212	0.259	0.497	0.315	0.265
8	0.116	0.220	0.270	0.201	0.198	0.090	0.212	0.000	0.132	0.476	0.217	0.196
9	0.116	0.180	0.225	0.169	0.164	0.095	0.259	0.132	0.000	0.399	0.122	0.116
10	0.458	0.384	0.495	0.381	0.378	0.397	0.497	0.476	0.399	0.000	0.307	0.336
11	0.148	0.127	0.270	0.138	0.156	0.114	0.315	0.217	0.122	0.307	0.000	0.079
12	0.127	0.138	0.241	0.111	0.130	0.127	0.265	0.196	0.116	0.336	0.079	0.000

ν	1	2	3	4	5	6	7	8	9	10	11	12
1	0.000	0.220	0.386	0.188	0.132	0.077	0.185	0.172	0.090	0.452	0.138	0.111
2	0.220	0.000	0.466	0.228	0.172	0.228	0.362	0.317	0.146	0.286	0.135	0.138
3	0.386	0.466	0.000	0.476	0.405	0.344	0.286	0.251	0.394	0.537	0.394	0.389
4	0.188	0.228	0.476	0.000	0.143	0.169	0.283	0.307	0.201	0.397	0.175	0.198
5	0.132	0.172	0.405	0.143	0.000	0.153	0.272	0.259	0.135	0.341	0.098	0.079
6	0.077	0.228	0.344	0.169	0.153	0.000	0.135	0.169	0.101	0.439	0.143	0.159
7	0.185	0.362	0.286	0.283	0.272	0.135	0.000	0.146	0.209	0.505	0.267	0.275
8	0.172	0.317	0.251	0.307	0.259	0.169	0.146	0.000	0.206	0.415	0.217	0.233
9	0.090	0.146	0.394	0.201	0.135	0.101	0.209	0.206	0.000	0.405	0.119	0.101
10	0.452	0.286	0.537	0.397	0.341	0.439	0.505	0.415	0.405	0.000	0.328	0.344
11	0.138	0.135	0.394	0.175	0.098	0.143	0.267	0.217	0.119	0.328	0.000	0.071
12	0.111	0.138	0.389	0.198	0.079	0.159	0.275	0.233	0.101	0.344	0.071	0.000

Table 3: Maximal and minimal values of positive and negative consonance between the twelve pillars of competitiveness for years 2008–2009 to 2013–2014.

Year	μ		ν	
	$\max(\mu_{C_iC_j})$	$\min(\mu_{C_iC_j})$	$\max(\nu_{C_iC_j})$	$\min(\nu_{C_iC_j})$
2008–2009	0.860	0.389	0.497	0.077
2009–2010	0.865	0.410	0.505	0.071
2010–2011	0.852	0.447	0.468	0.087
2011–2012	0.870	0.405	0.534	0.074
2012–2013	0.870	0.421	0.519	0.071
2013–2014	0.873	0.399	0.537	0.071

From Table 3, we can make certain conclusions about the range of values of the parameters α and β, which are used to measure the consonance between the criteria. Obviously, depending on how the values of α and β have been chosen, different sets of correlating criteria will form; and this can be done over the data for each year. For the purposes of illustration, let us only take the data for the latest period (2013–2014), and check how the relations between the criteria change by selecting different values of α and β. Obviously, in this case putting $\alpha > 0.873$ or $\beta < 0.071$ would yield no results.

In general, the question how to select the values of α and β, with respect to our various needs and purposes, is important and challenging, but is beyond the scope of the present research. Hence, we will conduct our analysis by taking the following exemplary pairs of (α; β): (0.85; 0.15), (0.80; 0.20), (0.75; 0.25), (0.70; 0.30), (0.65; 0.35), and will see which pillars are in positive consonance (Table 4, those in negative consonance follow by analogy).

Obviously, values $\alpha = 0.85$; $\beta = 0.15$ are rather discriminative, since only two consonance pairs are discovered to hold between four different criteria: 'Institutions – Technological readiness' and 'Business sophistication – Innovation', the second one being quite natural, since these two pillars take part in the formation of the 'Innovation and sophistication factors' defining the difference between the efficiency driven countries (2nd stage of development) and innovation driven countries (3rd stage of development). The rest two criteria are of more heterogeneous nature, where 'Institutions' belongs to the set of 'Basic requirements' and 'Technological readiness' belongs to the set of 'Efficiency enhancers'.

Table 4: List of pillars in positive consonance for the year 2013–2014, per different α, β. Highlighted in grey on each row are those consonances, which have been reported on previous (upper) rows, the white ones appearing for first.

(α, β)	List of positive consonances C_i–C_j	No. of μ-pairs	No. of ν-pairs	No. of consonances	No. of involved criteria
(0.85; 0.15)	1–9; 11–12	2	19	2	4
(0.80; 0.20)	1–5; 1–6; 1–9; 1–11; 1–12; 2–11; 5–11; 5–12; 9–11; 9–12; 11–12	11	29	11	7
(0.75; 0.25)	1–5; 1–6; 1–9; 1–11; 1–12; 2–9; 2–11; 2–12; 5–9; 5–11; 5–12; 6–9 6–11; 6–12; 9–11; 9–12; 11–12	17	37	17	7
(0.70; 0.30)	1–2; 1–4; 1–5; 1–6; 1–7; 1–8; 1–9; 1–11; 1–12; 2–5; 2–9; 2–11; 2–12; 4–5; 4–11; 5–6; 5–9; 5–11; 5–12; 6–7; 6–8; 6–9; 6–11; 6–12; 7–8; 8–9; 9–11; 9–12; 11–12	29	45	29	10
(0.65; 0.35)	1–2; 1–4; 1–5; 1–6; 1–7; 1–8; 1–9; 1–11; 1–12; 2–4; 2–5; 2–6; 2–9; 2–10; 2–11; 2–12; 3–8; 4–5; 4–6; 4–9; 4–11; 4–12; 5–6; 5–9; 5–11; 5–12; 6–7; 6–8; 6–9; 6–11; 6–12; 7–8; 7–9; 8–9; 8–11; 8–12; 9–11; 9–12; 11–12	39	51	39	12

The rest investigated values of α and β are looser, thus yielding greater number of consonance pairs between larger sets of criteria. We make the detailed analysis only for the second pair, (0.8; 0.2).

Putting $\alpha > 0.8$, we obtain 11 pairs of criteria which have their $\mu > 0.8$; and putting $\beta < 0.2$, we obtain 29 pairs of criteria which have their $\nu < 0.2$. The first set of 11 pairs is completely a subset of the second set of 29 pairs, meaning that we will discuss only these 11 pairs, which are in positive consonance; they connect 7 out of 12 pillars, as shown in Table 5.

Table 5: List of pillars in positive consonance for the year 2013–2014, when $\alpha > 0.8, \beta < 0.2$.

C_i–C_j	Full titles of criteria C_i–C_j	$\mu_{C_iC_j}$	$\nu_{C_iC_j}$
1–5	Institutions – Higher education and training	0.807	0.132
1–6	Institutions – Goods market efficiency	0.836	0.077
1–9	Institutions – Technological readiness	0.854	0.090
1–11	Institutions – Business sophistication	0.804	0.138
1–12	Institutions – Innovation	0.844	0.111
2–11	Infrastructure – Business sophistication	0.804	0.135
5–11	Higher education and training – Business sophistication	0.815	0.098
5–12	Higher education and training – Innovation	0.847	0.079
9–11	Technological readiness – Business sophistication	0.810	0.119
9–12	Technological readiness – Innovation	0.831	0.101
11–12	Business sophistication – Innovation	0.873	0.071

Putting $\alpha = 0.75$; $\beta = 0.25$, we obtain 17 pairs w.r.t. α and 37 pairs w.r.t. β, giving a total of 17 pairs of consonance w.r.t. both parameters at a time. In these 17 pairs take part again the same 7 criteria, as in the previous case (0.80; 0.20), but 6 more correlations between them are now discovered, namely, 'Infrastructure – Technological readiness', 'Infrastructure – Innovation', 'Higher education and training – Technological readiness', 'Goods market efficiency – Technological readiness', 'Goods market efficiency – Business sophistication' and 'Goods market efficiency – Innovation'.

The pairs (0.70; 0.30) and (0.65; 0.35) are rather inclusive and non-discriminative values, since they involve, respectively, 10 and 12 out of 12 pillars of competitiveness and yield, respectively, 29 and 39 correlations between them.

We can visually illustrate the findings in Table 4 by constructing graphs for each run of α and β, depicting the outlined dependences. We will do it here only for the described case when $\alpha > 0.8$, $\beta < 0.2$, see Figure 1.

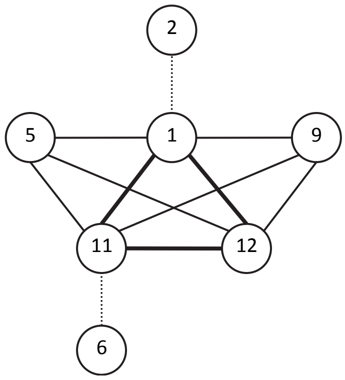


Figure 1: Graph structure of the pillars forming positive consonance for the year 2013–2014 when $\alpha > 0.8, \beta < 0.2$.

Now it becomes rather visual that when $\alpha > 0.8$, $\beta < 0.2$ three out of seven pillars completely correlate with each other ('1. Institutions', '11. Business sophistication', '12. Innovation'), two other ('5. Higher education and training' and '9. Technological readiness') completely correlate with the triple 1–11–12, but not among each other, while vertices '2. Infrastructure' and '6. Good market efficiency' are connected by only one arc to the rest of the structure.

Obviously, for each run of α and β a series of graphs will be formed, where every consequent graph will act as a supergraph for the previous one, becoming gradually more complex and intercomnected. It is interesting to compare for each run of α and β whether and how these graph structures change over the different time periods before 2013–2014.

These graph structures are a matter of further economic analysis, and it is particularly interesting to study which of the pillars of competitiveness are fully connected, like 1–5–11–12 and 1–9–11–12 in Figure 1.

Also, it is noteworthy that in the WEF's methodology for forming the countries' competitiveness index, there are four sub-indicators take part in two pillars each, namely: 'Intellectual property protection' takes part of the formation of the 1st and 12th pillar, 'Mobile telephone subscriptions' and 'Fixed telephone lines' in 2nd and 9th pillar, and 'Reliance on professional management' in 7th and 11th pillar.

We can hence make the conclusion, that our findings generally support the proximity between the mentioned pillars, as suggested by the presence of shared sub-indicators, yet our conclusions are much stronger and sophisticated as a result of the research.

It is also very important to make the comparison of the calculated values in Tables 1 and 2 between years 2008–2009 and 2013–2014. We can focus the reader's attention to several particularly well outlined observations. Over the period 2008–2014, the pillars '5. Higher education and training' and '7. Labour market efficiency' have become gradually more correlated to all the rest pillars, while pillar '3. Macroeconomic stability' has become gradually less correlated. However, in general, these comparisons are a matter of detailed analysis by economists.

4 CONCLUSION

The present research aimed at discovery of some hidden patterns in the data about EU Member States' competitiveness in the period from 2008 to 2014. We conduct the analysis of the World Economic Forum's Global Competitiveness Reports, using a recently developed multicriteria decision making method, based on index matrices and intuitionistic fuzzy sets.

Using index matrices with data about how the EU Member States have performed according to the outlined twelve 'pillars of competitiveness', we construct new matrices, giving us new knowledge about how these pillars correlate and interact with each other. Moreover, the application of the method has been traced over a six-year period of time and has revealed certain changes and trends in these correlations that may yield fruitful further analyses by interested economists. The results are illustrated with data tables and graphs of the strongest correlations between the criteria.

These conclusions may also be useful for the national policy and decision makers, to better identify and strengthen the transformative forces that will drive their future economic growth. The same approach can be equally applied to other selections of countries and time periods, and comparisons with the hitherto presented results will be challenging.

Besides the comparison of the twelve pillars of competitiveness, our research plans include also exploring the correlations between the most problematic factors for doing business, as outlined in the WEF's GCRs. Further investigation how the pillars of competitiveness correlate with these most problematic factors may also prove interesting and useful.

ACKNOWLEDGEMENTS

The research work reported in the paper is partly supported by the project AComIn "Advanced Computing for Innovation", grant 316087, funded by the FP7 Capacity Programme (Research Potential of Convergence Regions).

REFERENCES

Atanassov K. (1983) Intuitionistic fuzzy sets, VII ITKR's Session, Sofia, June 1983 (in Bulgarian).

Atanassov K. (1986) Intuitionistic fuzzy sets. Fuzzy Sets and Systems. Vol. 20 (1), pp. 87–96.

Atanassov K. (1991) Generalized Nets. World Scientific, Singapore.

Atanassov K. (1999) Intuitionistic Fuzzy Sets: Theory and Applications. Physica-Verlag, Heidelberg.

Atanassov K. (2012) On Intuitionistic Fuzzy Sets Theory. Springer, Berlin.

Atanassov K., D. Mavrov, V. Atanassova (2013). Intercriteria decision making. A new approach for multicriteria decision making, based on index matrices and intuitionistic fuzzy sets. Proc. of 12th International Workshop on Intuitionistic Fuzzy Sets and Generalized Nets, 11 Oct. 2013, Warsaw, Poland (in press).

World Economic Forum (2008, 2013). The Global Competitiveness Reports. http://www.weforum.org/issues/global-competitiveness.

Zadeh L. A. (1965). Fuzzy Sets. Information and Control Vol. 8, pp. 333–353.

INDUSTRIAL TRACK

Value-driven Design and Implementation of Business Processes

Transferring Strategy into Execution at Pace with Certainty

Mathias Kirchmer[1,2]
[1] *BPM-D, 475 Timberline Trail, West Chester, PA 19382, U.S.A.*
[2] *Organizational Dynamics, University of Pennsylvania, Philadelphia, U.S.A.*
mathias.kirchmer@bpm-d.com

Keywords: ARIS, BPM, BPM-discipline, Execution, Innovation, Optimization, Modelling, Process Design, Process Implementation, Reference Models, Strategy, Value-Driven BPM.

Abstract: An organization only competes with approximately 20% of its business processes. 80% of the processes are commodity processes that can have an industry average performance. A value-driven process design and implementation considers this by focusing innovation and optimization initiatives as well as individual software development on the 20% high impact processes, while commodity processes are designed based on industry reference models and implemented as far as possible through standard software. The paper describes an approach to such a value-driven design and implementation of businesses – transferring strategy into execution, at pace with certainty.

1 INTRODUCTION

A key challenge of organizations in today's volatile business environment is "leveraging people to build a customer-centric performance-based culture" (Mitchel, Ray, van Ark, 2014). Therefore it is not only important to have a good strategy, hence to know what to do. But in many organizations the key challenge is about how to do things in order to build such a customer and market-oriented organization. This can be achieved through a consequent process-orientation since processes deliver by definition a result of value for a client outside the process. However, this requires a structured value-driven design of processes focusing on realizing the business strategy of an organization (Rummler, Ramias, Rummler, 2010) (Burlton, 2010).

This paper presents such a value-driven process design and implementation approach that is both, focused on executing the strategy of an organization while being as resource efficient as possible. Result is a practical and effective approach to process design and implementation.

The approach has been developed based on practical experience in large and mid-size organizations, mainly in the USA and Europe. It has been combined with academic research regarding such value-driven design and implementation methodologies.

2 TARGETING VALUE

The overall goal of the approach is to target and focus on business value during the design and implementation of processes. Research has shown that organizations only compete with approximately 20% of their business processes (Franz, Kirchmer, 2012). This means that 80% of the business processes are commodity processes which can be carried out according to industry standards or common industry practices. An average industry performance is sufficient. Sophisticated improvement approaches targeting higher performance are not delivering real additional busienss value. Hence, process innovation and optimization initiatives have to focus on the 20% high impact processes while other business processes can be designed and implemented using existing industry common practices. Results are highly organization specific business processes where this really delivers competitive advantages and processes following industry common practices where this is sufficient.

Such an approach also enables organizations to use resources where they provide best value during design and implementation initiatives. People who are highly qualified in sophisticated process design and implementation methods, for example, focus on high value areas. They can systematically target value as well as reduce the risk of project failure (Kirchmer 2013). They can focus on moving the organization to the next level. This reflects requirements of a modern Chief Information Officer (CIO) (Scheer 2013) who moves away from being a technical expert to a driver of innovation and performance. Hence, the approach allows such a person to transition into a Chief Process Officer (Franz, Kirchmer, 2012).

Such an approach requires the appropriate segmentation of processes as basis for a differentiated design and implementation approach. Process models developed during the process design need to reflect the requirements of those different process segments and the importance of the resulting business processes for the strategy of an organization. Different levels of sophistication regarding the improvement approaches are necessary.

The following process implementation, including the appropriate software support, is executed accordingly to the process design based on the identified process segments. The value-driven design leads often to different approaches to procure the enabling software. Highly organization-specific processes often require an individual development of software. Processes designed based on industry standards lead in most cases to the use of standard software packages.

Such a value-driven process design and implementation approach enables the required performance of an organization while meeting efficiency goals. It helps creating a culture with the right performance and customer focus by moving strategy into execution at pace with certainty.

3 SEGMENTING PROCESSES

A business process assessment based on the impact of a business process on main strategic value-drivers is the basis for the segmentation of processes into high impact and commodity processes (Franz, Kirchmer, 2014). This process assessment is the key tool to align business strategy with process design and implementation. It enables the desired value-driven approach and makes it a management discipline to transfer business strategy into execution.

The value-drivers are deducted from the business strategy of the organization using value-driver-tree models (value-driver trees). This is a way of transferring the strategic intension of an organization into operational value-driven targets. An example for such a value-driver tree is shown in figure 1. The value-drivers themselves can again be prioritized to focus the segmentation on the most important value-drivers.

In practice a three step approach has proven to be successful. The strategy delivers the business priorities which are decomposed into strategic objectives. Then one or several value-drivers are identified for each objective, hence the drivers that make this objective happen.

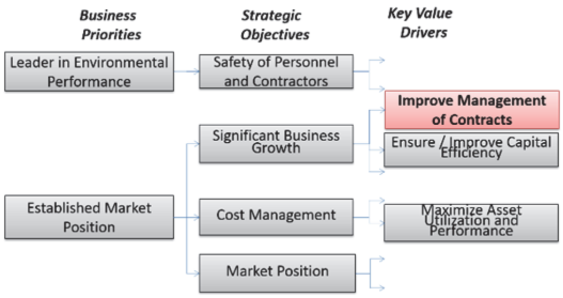

Business Priorities
Strategic Objectives
Key Value Drivers
Leader in Environmental Performance
Safety of Personnel and Contractors
Improve Management of Contracts
Significant Business Growth
Ensure / Improve Capital Efficiency
Established Market Position
Cost Management
Maximize Asset Utilization and Performance
Market Position

Figure 1: Value-driver tree (Excerpt).

The business processes of an organization are then evaluated based on their total impact on the specific value-drivers. Result are two segments of business processes: high impact and commodity processes. "High impact" processes are the ones that are key to make the business strategy of the organization happen. They link strategy to execution. This approach is visualized in figure 2.

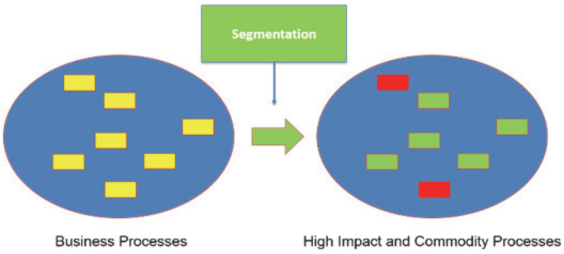
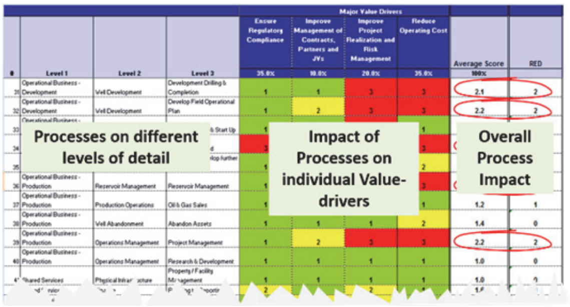


Figure 2: High impact and commodity processes.

The value-drivers can be weighted regarding their importance. Minor changes and adjustments in strategy can then be just reflected through adjustments of those weights. Larger strategy changes result in different or additional value-drivers. For each process it has to be defined if it has no (0), low (1), medium (2) or high (3) impact on each of the value-drivers. Then the overall impact is calculated in a process assessment matrix by multiplying impact with the weight of the appropriate value-driver and calculating the total of all impacts of a process. An example of a process assessment matrix is shown in figure 3.


Processes on different levels of detail
Impact of Processes on individual Value-drivers
Overall Process Impact

				Major Value Drivers					
#	Level 1	Level 2	Level 3	Ensure Regulatory Compliance	Improve Management of Contracts, Partners and JVs	Improve Project Realization and Risk Management	Reduce Operating Cost	Average Score	RED
				35.0%	10.0%	20.0%	35.0%	100%	
31	Operational Business - Development	Well Development	Development Drilling & Completion	1	1	3	3	2.1	2
32	Operational Business - Development	Well Development	Develop Field Operational Plan	1	2	3	3	2.2	2
33	Operational Business -	[illegible]	& Start Up	1	[illegible]	[illegible]	1	[illegible]	[illegible]
34	[illegible]	[illegible]	d	3	[illegible]	[illegible]	3	[illegible]	[illegible]
35	[illegible]	[illegible]	lop further	1	[illegible]	[illegible]	2	[illegible]	[illegible]
36	Operational Business - Production	Reservoir Management	Reservoir Management	1	[illegible]	[illegible]	3	[illegible]	[illegible]
37	Operational Business - Production	Production Operations	Oil & Gas Sales	1	[illegible]	[illegible]	1	1.2	1
38	Operational Business - Production	Well Abandonment	Abandon Assets	1	1	1	2	1.4	0
39	Operational Business - Production	Operations Management	Project Management	1	2	3	3	2.2	2
40	Operational Business - Production	Operations Management	Research & Development	1	1	1	1	1.0	0
41	Shared Services	Physical Infrastructure	Property / Facility Management	1	1	1	1	1.0	0
[illegible]	[illegible]	[illegible]	[illegible]	2	[illegible]	2	[illegible]	[illegible]	[illegible]

Figure 3: Process assessment matrix (Excerpt).

The high impact processes have then to be evaluated based on general industry practices, e.g. through benchmarks or purely qualitative evaluations. In that way you identify the high impact "high opportunity" business processes. These are the processes where improvements have the biggest value potential since the process has a high impact on the strategy but it currently performs only in or even under the industry average.

Practice experience with different companies has shown that the processes should be identified on a level of detail so that 150-200 process definitions describe the entire organization. This is often referred to as "level 3" (L3). This level is detailed enough to obtain differentiated results but high level enough to avoid to high work efforts. Using the results of the process assessment matrix the 20% of the processes that are classified as high impact can be identified. The others are the commodity processes.

In practice there is often a "grey" area of processes that could be in either group. Hence there may be slightly more or less than 20% of the processes in the high impact segment. This issue has to be resolved in a case to case basis reflecting the specific situation of an organization and its business strategy.

4 VALUE-DRIVEN DESIGN

The high impact processes (or at least high impact high opportunity processes, if further prioritization is necessary, e.g. due to budgets) are subject to detailed process innovation and optimization activities focusing on the previously identified value-drivers (Kirchmer, 2011). Therefore product and market-oriented design approaches (Kirchmer, 1999b) have been proven effective. The approach is used in conjunction with the application of standard modelling methods like Event-driven Process Chains (EPC) or the Business Process Modelling Notation (BPMN) to facilitate the integration of process design and implementation. The product and market-oriented design supports an integrated product (offering) and process innovation. Such an approach is especially important for the processes that are highly relevant for the strategic positioning of an organization. In order to identify these business processes another segmentation of the high impact processes is required distinguishing between strategic and non-strategic high impact processes. The focus is on high impact strategic processes (Franz, Kirchmer, 2012). These are perfect targets for innovation initiatives. As an example, a compressor company may deliver "compressed air as a service" instead of just selling compressors. Offering and related sales processes change simultaneously – reflected in the integrated design.

For all high impact processes techniques like process model based simulations and animations are helpful to come up with best suited design solutions. Traditional improvement methods like Lean or Six Sigma (George, 2010) can be applied in selected cases. However, these are in general not approaches that support a focused innovation. Hence, they are more targeted to bringing less strategic people intense processes to better efficiency, in most cases resulting in cost or time reductions.

A structured modelling and design approach is essential to produce models that enable a seamless link to implementation. A perfect framework is "ARIS", the Architecture of Integrated Information Systems (Scheer 1998). This facilitates the design of processes from different view points: organization, functions, data, deliverables and control flow. Result are process models that contain all information for the following implementation. The ARIS architecture is shown in figure 4.

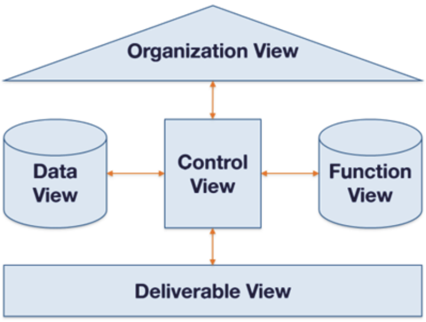


Figure 4: ARIS architecture by A.-W. Scheer.

Starting point for the design of commodity processes are industry reference models. These models are available for example through industry organizations, consulting and software companies

(Kirchmer, 2011). In many cases they are already developed using standard modelling methods. The industry common practices reflected in those models are only adjusted to the specific organization when this is absolutely necessary. The process design work focusses on "making the industry standard happen". If process areas are identified where the industry standard cannot be applied, e.g. due to product specifics, only those areas will be designed in a company specific way, keeping the adjustments as close to the industry standard as possible. Process solutions can here often be found through a simple application of the mentioned traditional improvement methods like Lean and Six Sigma since a pure efficiency focus is in most cases justified here. However, it is important to keep in mind that it is in general not worth improving above industry standard performance.

This value-driven process design approach is visualized in figure 5. It shows that also for the design of high impact processes reference models can be used as an input. But this is only one component of getting all information together to come up with real innovative and optimized solutions.

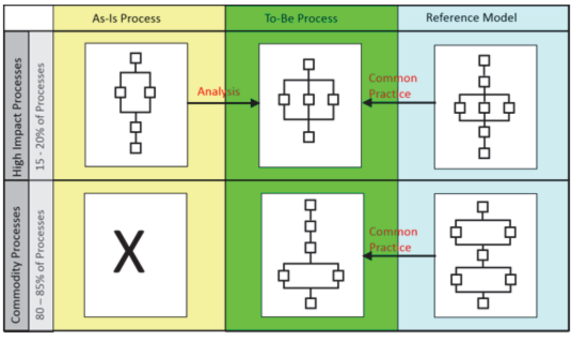


Figure 5: Value-driven process design approach.

In both cases process models are developed until the level of detail that still provides relevant business information through the design. The decomposition of the function "Enter Customer Order" into "Enter First Name", "Enter Last Name", etc. would from a business point of view not add any additional relevant content (but may be necessary later for the development of software). When reference models are used this can mean that in areas where the design deviates from the initial industry model a higher level of modelling detail is required then in other "standard" areas.

Both, high impact and commodity processes are part of overlying end-to-end business processes. Process-interfaces in the underlying detailed processes reflect this overall context and make sure that the various process components or sub-processes fit together. Hence, during the process improvement work cause-and-effect considerations have to take place in order to avoid fixing issues in one area while creating new ones in other processes.

5 VALUE-DRIVEN IMPLEMENTATION

The very organization specific process models for high impact business processes are in general implemented using highly flexible next generation process automation engines and require in most cases the development of application software components. The process models reflecting the optimized and innovative design are the entrance point for the more detailed modelling of the underlying software. At this point the modelling method can change, for example to the Unified Modelling Language (UML), reflecting the desired software structure to support the high impact processes. Also the workflow engine of next generation process automation engines can be configured based on those models, depending on the underlying modelling and execution technology even automatically or semi-automatically. The integration between process modelling and execution tools can be extremely beneficial in this situation.

The overall architecture of such next generation process automation environments is often referred to as Service Oriented Architecture (SOA). In such an architecture the "execution software" and the "process logic" (workflow) are separated (Kirchmer, 2011) (Slama, Nelius, 2011). Hence, the developed process models can on one hand be sued to configure the workflow and on the other hand to develop the software services that are not available in existing libraries. Existing software services may include detailed process reference models that can be included in the process design. This architecture of next generation process automation environments is visualized in figure 6.

Key advantage of such an architecture is the high degree of flexibility in adjusting process flows and functionality. This can be crucial for a company looking for agility and adaptability. Main disadvantage is the effort for providing the appropriate governance while running such an environment as well as modelling efforts in the building phase.

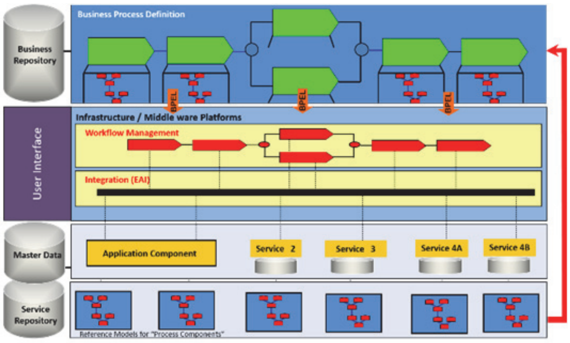


Figure 6: Next generation process automation.

The process models of the commodity processes are used to select or at least evaluate pre-selected "traditional" software packages like Enterprise Resource Planning (ERP) systems, Supply Chain Management (SCM) or Customer Relationship Management (CRM) systems. These can become part of the overall next generation architecture, representing one software component. Then those models from the process design are used to drive a process-oriented implementation of the software packages across the various organizational units involved in the business processes in scope (Kirchmer, 1999). Ideally one uses already industry specific software-reference models during the process design. This means, one procures the reference models to be used from the software vendor. Hence, one benefits from the "business content" of the software and minimizes design and modelling efforts. Using other industry reference models (different from the software based model) may lead to design adjustments once the software is selected.

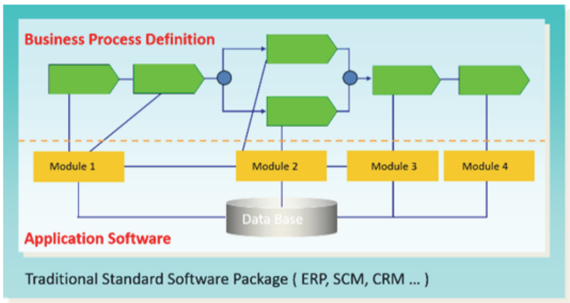

Business Process Definition
Module 1
Module 2
Module 3
Module 4
Data Base
Application Software
Traditional Standard Software Package (ERP, SCM, CRM ...)

Figure 7: Traditional software architecture.

Figure 7 shows the architecture of such a traditional software. Here process definition and software functionality are linked in a static way. This means the software more or less dictates how a process has to be executed (allowing only pre-defined variants through the software configuration). This is fine for commodity processes but causes issues in strategic high impact processes that need to be company specific. Consequently we have used another implementation approach there. However, in some cases it is also possible to develop add-on software to support high impact processes and integrate it into the larger software package, e.g. the ERP system.

Advantages and disadvantages are just the opposite as explained for next generation process automation approaches. Hence, in practice a combination of both efforts is in most cases the solutions that delivers best value.

The process-interfaces in the different process models guide the software integration. This can be supported from a technology point of view through appropriate enterprise application integration environments – in general included in SOA environments. Such software tools or middle-ware tools reduce the efforts for interface development to a necessary minimum. Their efficient use is again driven through the appropriate process models, specifically the integration of the various process components.

Result are end-to-end business processes based on a value-driven process design and an appropriate integrated automation. The approaches provides the necessary flexibility where it delivers real business value and the required efficiency where possible. Business process management, software selection and development are integrated in one overlaying discipline of value-driven business process management (BPM) (Franz, Kirchmer, 2014) (Franz, Kirchmer, 2012).

6 CONCLUSIONS

The approach of value-driven business process design and implementation allows an organization to move its strategy systematically into execution. It aligns the modelling and implementation efforts with the strategic direction of the organization.

First experiences with real live companies showed that this approach helps on one hand to dramatically reduce process design and implementation times due to the efficient handling of commodity processes. Companies estimated more than 50% savings in time and effort. On the other hand it enables real strategic advantage though the innovation and optimisation of high impact process areas.

While the basic approach has proven to be successful in practice there are still gaps to close. In the design part the systematic achievement of appropriate process innovation is still a topic that needs further research. Considering the importance of process innovation, this is a real key topic. In the field of process implementation the integration of the process modelling and execution environments can still be improved. While there is quite a bit of progress on the software-side (Scheer 2013) (Stary 2012), there is still work to do on an integrating software and organizational governance solution.

REFERENCES

Burlton, 2013. Delivering Business Strategy through Process Management. In: Vom Brocke, J., Roemann, M.: Handbook on Busienss Process Management 2 – Strategic Aignment, Governance, People and Culture, Springer, Berline, New York, e.a.

Elzina, Gulledge, Lee, 1999. Business Engineering, Springer, Norwell.

Franz, Kirchmer, 2012. Value-driven business Process Management – The Value-Switch for Lasting Competitive Advantage, McGraw-Hill, New York, e.a.

Franz, Kirchmer, 2014. The BPM-Discipline – Enabling the Next Generation Enterprise. BPM-D Executive Training Documentation, London, Philadelphia.

George, 2010. The Lean Six Sigma Guide to Doing More with Less – Cut Costs, Reduce Waste, and Lower your Overhead. McGraw-Hill, New York, e.a.

Kirchmer, 2013. How to create successful IT Projects with Value-driven BPM, In: CIO Magazine Online, February 27th 2013.

Kirchmer, 2011. High Performance through Process Excellence – From Strategy to Execution with Business Process Management. Springer, 2nd edition, Berlin, e.a.

Kirchmer, 1999a. Business Process Oriented Implementation of Standard Software – How to achieve Competitive Advantage Efficiently and Effectively. Springer, 2nd edition, Berline, e.a.

Kirchmer, 1999b. Market- and Product-oriented Definition of Business Processes. In: Elzina, D.J., Gulledge, T.R., Lee, C.-Y (editors): Business Engineering, Springer, Norwell.

Mitchel, Ray., van Ark, 2014.: The Conference Board – CEO Challenge 2014: People and Performance, Reconnecting with the Customer and Reshaping the Culture of Work. The Conference Board Whitepaer, New York, e.a. 2014.

Rummler, Ramias, Rummler, 2010: White Space Revisited – Vreating Value thorugh Processes.Wiley, San Francisco.

Scheer, 2013. Tipps fuer den CIO: Vom Tekki zum Treiber neuer Business Modelle. In: IM+IO – Das Magazin fuer Innovation, Organisation und Management, Sonderausgabe, Dezember 2013.

Scheer, 1998. ARIS – Business Process Frameworks, Springer, 2nd edition, Berlin, e.g.

Slama, Nelius, 2011. Enterprise BPM – Erfolgsrezepte fuer unternehmensweites Prozessmanagement. dpunkt.verlag, Heidelberg.

Stary, 2012. S-BPM One – Scientific Research. 4th International Conference, S-BPM ONE 2012, Vienna, Austria, April 2012, Proceedings.

AUTHOR INDEX

Proceedings of BMSD 2014
Fourth International Symposium on Business Modeling and Software Design
ISBN: 978-989-758-032-1
http://www.is-bmsd.org