Xavier Franch
Pnina Soffer (Eds.)

Advanced Information Systems Engineering Workshops

LNBIP 148

CAiSE 2013 International Workshops
Valencia, Spain, June 2013
Proceedings

Springer

Lecture Notes
in Business Information Processing 148

Series Editors

Xavier Franch
Pnina Soffer (Eds.)

Advanced Information Systems Engineering Workshops

CAiSE 2013 International Workshops
Valencia, Spain, June 17-21, 2013
Proceedings

Springer

Volume Editors

Xavier Franch
Universitat Politècnica de Catalunya
Service and Information System Engineering
Barcelona, Spain
E-mail: franch@essi.upc.edu

Pnina Soffer
University of Haifa
Information Systems
Haifa, Israel
E-mail: spnina@is.haifa.ac.il

ISSN 1865-1348 e-ISSN 1865-1356
ISBN 978-3-642-38489-9 e-ISBN 978-3-642-38490-5
DOI 10.1007/978-3-642-38490-5
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2013938414

ACM Computing Classification (1998): H.4, H.3.5, D.2, J.1, I.2

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

These proceedings include the papers from workshops held in conjunction with the 25th Conference on Advanced Information Systems Engineering (CAiSE 2013) in Valencia, Spain. The CAiSE conference is a leading and prestigious event, where state-of-the-art results in the area of information systems engineering are presented and discussed. In addition to the main conference, the CAiSE series of conferences has a long tradition of hosting workshops.

The workshops complement the conference and provide forums for researchers and practitioners to exchange ideas and share initial results in an atmosphere that fosters interaction. Each workshop has a relatively narrow focus, facilitating lively discussions among participants.

The CAiSE 2013 workshops were selected based on quality, relevance, and reputation. Some of the workshops have a successful history with CAiSE, while others are held for the first time. All the workshops whose papers appear in this volume have undertaken upon themselves to maintain a high-quality and selective acceptance policy, resulting in acceptance rates of up to 50% for full research papers.

The workshops whose papers appear in these proceedings are:

- Approaches for Enterprise Engineering Research (AppEER)
- International Workshop on BUSiness/IT ALignment and Interoperability (BUSITAL)
- International Workshop on Cognitive Aspects of Information Systems Engineering (COGNISE)
- Workshop on Human-Centric Information Systems (HC-IS)
- Next Generation Enterprise and Business Innovation Systems (NGEBIS)
- International Workshop on Ontologies and Conceptual Modeling (OntoCom)
- International Workshop on Variability Support in Information Systems (VarIS)
- International Workshop on Information Systems Security Engineering (WISSE)

Other workshops that were held during CAiSE, whose papers are not contained in this volume, include International Workshop on Enterprise & Organizational Modeling And Simulation (EOMAS), and 6th International i* Workshop (iStar).

We would like to express our gratitude to the workshop organizers who initiated the workshops and took responsibility for the programs of their workshop.

They managed the process from issuing the call for papers, reviewing and selecting the papers, and preparing their part of the proceedings. We would also like to thank the members of the various Program Committees who devoted their time and helped us put together an exciting workshop program at CAiSE 2013. Finally, we are thankful to the CAiSE 2013 organizers for all their efforts to make this a successful, exciting, and enjoyable event.

June 2013

Xavier Franch
Pnina Soffer

Approaches for Enterprise Engineering Research Workshop AppEER

The AppEER Workshop (Approaches for Enterprise Engineering Research)[1] is set up as a one-day event in such a way that it attracts researchers (academics and practitioners). AppEER 2013 was focused on advancing information systems research and was co-located with the 25th International Conference on Advanced Information Systems Engineering (CAiSE 2013) in Valencia, Spain.

Information systems are used in enterprises to support human actors in their activities. Essentially, information systems amplify human information processing capabilities. When engineering such information systems, one must therefore do so in full alignment with their human/organizational contexts. This involves the coherent *engineering* of a socio technical system. Studying the engineering of such socio technical systems has resulted in a wide range of disciplines, including business process management, enterprise architecture, enterprise modeling, enterprise transformation, business engineering, organizational engineering, etc. As the overarching term we will use "enterprise engineering" (EE).

The research field of EE takes an engineering perspective on the design and establishment of enterprises and their supportive (information) technologies. This requires insights into a broad range of aspects, such as human, cultural, organizational, and political aspects of enterprises and the processes involved in establishing them. This broad range of aspects touches on various existing research fields, including organizational science, management science, and information science, that form the "source" fields on EE.

The multidisciplinarity of the EE field also implies the need to consider it from differing (potentially contrasting) points of view. As a result, various research approaches, research methodologies are needed to expand and deepen the EE research field in terms of theories, models, methods, and other instruments for the analysis, design, implementation, evolution, and governance of enterprises.

In practice, researchers (academics or practitioners) face many challenges in selecting/synthesizing the most appropriate approach for doing EE research. Even more, being an emerging multidisciplinary field involves the challenge of establishing an open and multifaceted research tradition, drawing upon the strengths of the research fields it touches on. This multifaceted research tradition has not been established enough yet. This is especially needed, since the research approaches used in the different source fields have differing points of views on how to conduct research, potentially leading to clashes between these traditions.

[1] http://appeer.ee-team.eu/

The AppEER workshop intends to provide a common platform for the discussion and presentation of original work describing the usage of different approaches such as: design science, interpretivism, positivism, criticism, etc., for doing EE research. Moreover, the intended audience should involve people who have the ambition to bring "engineering rigour" to the design/development/creation of enterprises and their information systems. Basically, how organizations currently deal/grapple with the task of designing/evolving themselves and their ISs; how they currently use engineering approaches in this; and how the design of engineering methods/techniques/approaches is used to do/improve EE.

As a first event, this workshop succeeded in attracting different communities from research fields such as organizational science, management science and information science composing the "source" fields on EE. The workshop attracted nine submissions. The submitted papers came from both academics and practitioners who shared their original insights concerning research approaches in EE. Every paper received more than three reviews and was independently discussed with the advisory board. In the end, we decided to accept five papers and agreed on adding an "invited paper" that fits with the goals of this workshop. Paper presentations are opportunities for stimulating meaningful discussions between participants, with the goal of developing approaches for research methodologies in EE, thereby creating synergies and jointly identifying topics for further research in future AppEER events.

Organization

Workshop Co-chairs

Wolfgang Molnar	CRP Tudor, Luxembourg
Khaled Gaaloul	CRP Tudor, Luxembourg
Robert Winter	University of St. Gallen, Switzerland
Roel Wieringa	University of Twente, The Netherlands

Advisory Board

Henderik A. Proper	CRP Tudor, Luxembourg and Radboud University Nijmegen, The Netherlands
Michael Rosemann	Queensland University of Technology, Brisbane, Australia
Ulrich Frank	Duisburg-Essen University, Germany

Program Committee

Achmad Nizar Hidayanto	Faculty of Computer Science University of Indonesia
David Aveiro	Madeira University
Dirk Vanderlinden	CRP Tudor Luxembourg
Gayuh Prima	Fasilkom UI
François Charoy	Université de Lorraine - LORIA - Inria
Frank Harmsen	Ernst & Young Advisory
Hans Mulder	Universiteit Antwerpen
Jan Verelst	University of Antwerp
João Pombinho	Instituto Superior Técnico - Technical University of Lisbon
Jean-Sébastien Sottet	CRP Tudor Luxembourg
Marion Lepmets	University of Dublin
Nils Labusch	University of St. Gallen
Pedro Malta	Minho University
Philip Huysmans	University of Antwerp
Ralf Abraham	University of St. Gallen
Robert Winter	University of St. Gallen
Sérgio Guerreiro	Universidade Lusófona de Humanidades e Tecnologias, Escola de Comunicação, Artes
Stijn Hoppenbrouwers	HAN University of Applied Sciences
Stephan Aier	University of St. Gallen
Sybren De kinderen	CRP Tudor Luxembourg
Wided Guedria	CRP Tudor Luxembourg

8th International Workshop on BUsiness/IT ALignment and Interoperability BUSITAL

Nowadays, information services are a core asset for organizations seeking sustainable business and competitive advantage. The continuous growth of information and communication technologies-enabled innovations provide organizations with new efficient mechanisms for communication, information sharing, resource management and planning, and help them to explore new market opportunities. The special theme of the 25th edition of CAiSE was "Information Services," where the notion of service plays an increasingly extensive role in enterprise development. However, in the context of enterprise development the notion of services should not be limited to information services. Accordingly, we stress the Service Science Research Manifesto by Chesbrough and Spohrer, which calls for an integrated view, namely, integrating management science with computer science. It is interesting that in this context the notion of service refers to different definitions: in management science, a service is defined as a business economic activity, offered by one party to another to achieve a certain benefit, and generated by business processes; in information systems, a service is a complex (or simple) task executed (within) by an organization on behalf of a customer; and in computer science, a service is a programmable, self-describing, encapsulated, and loosely coupled function accessed and invoked over the Internet.

As a consequence, the governance of information services in this increasingly evolving scenario demands new models of alignment not only within the traditional organizational boundaries, but also with an outer context that challenges organizations to anticipate the constantly evolving business, technological, and social environment, where interoperability issues are key success factors. Information systems have to support these evolutionary challenges while preserving the alignment between business strategies, business processes, social context, and application portfolios. Furthermore, recent decades have witnessed yet another wave of ICT innovations: cloud, smart and mobile technologies are rapidly integrating with our daily life and opening new business opportunities for organizations. Following new trends while mastering the complexity and gaining the maximum value from IT is the major challenge for both business and IT leaders. Traditionally, methods, approaches, theories, and applications of business-IT alignment have been vividly discussed by practitioners and researchers in IT. This 8th edition of the BUSITAL workshop clearly demonstrates the increasing interest on business-IT alignment from the management community.

April 2013

Christian Huemer
Irina Rychkova
Gianluigi Viscusi
Jelena Zdravkovic

Organization

BUSITAL 2013 was organized in conjunction with the 25th International Conference on Advanced Information Systems Engineering (CAiSE13) in Valencia, Spain.

Organizing Chairs

Christian Huemer	Vienna University of Technology, Austria
Irina Rychkova	University of Paris 1 Panthéon - Sorbonne, France
Gianluigi Viscusi	University of Milano-Bicocca, Italy
Jelena Zdravkovic	Stockholm University, Sweden

Program Committee

Joseph Barjis
Judith Barrios Albornoz
Alessio Maria Braccini
Eric Dubois
Pascal van Eck
Jaap Gordijn
Tharaka Ilayperuma
Paul Johannesson
John Krogstie
Kalle Lyytinen
Raimundas Matulevicius
Selmin Nurcan
Andreas L. Opdahl
Michal Petit
Yves Pigneur
Jolita Ralyte
Guttorm Sindre
Janis Stirna
Alain Wegmann
Hans Weigand
Roel Wieringa
Eric Yu
Jelena Zdravkovic

First International Workshop on Cognitive Aspects of Information Systems Engineering COGNISE

Cognitive aspects of information systems engineering is an area that is gaining interest and importance in industry and research. In recent years, human aspects and specifically cognitive aspects in software engineering and information systems engineering have received increasing attention in the literature and at conferences, acknowledging that these aspects are as important as the technical ones, which have traditionally been at the center of attention. This workshop was planned to be a stage for new research and vivid discussions involving both academics and practitioners.

The goal of this workshop is to provide a better understanding and more appropriate support of the cognitive processes and challenges practitioners experience when performing information systems development activities. Understanding the challenges and needs, providing educational programs, as well as developing supporting tools and notations may be enhanced for a better fit to our natural cognition, leading to better performance of engineers and higher system quality. The workshop aimed to bring together researchers from different communities, such as requirements engineering, software architecture, design and programming, who share an interest in cognitive aspects, for identifying the cognitive challenges in the diverse development-related activities.

The first edition of this workshop attracted six international submissions: two full papers and four position papers. Each paper was reviewed by several members of the Program Committee. From these submissions, one out of the two full papers was accepted and, in addition, all four submissions of position papers were accepted as short papers for presentation at the workshop.

The papers presented at the workshop provide a mix of novel research ideas, mainly presenting research in progress or research plans. The full research paper "Zooming In and Out in Requirements Engineering" by Manuel Imaz argues that user stories or use cases play an essential role in our cognition and are the basic-level categories in requirements engineering, which can be aggregated to more abstract components in a bottom-up manner, unlike traditional top-down approaches. The position paper "Cognitive Principles to Support Information Requirements Agility" by Jeffrey Parsons and Yair Wand proposes information systems design rules, based on classification theory to foster information requirements agility by decoupling a conceptual view of data from logical models. "Naomi Unkelo-Spigel and Irit Hadar" present in their paper "Using Distributed Cognition Theory for Analyzing the Deployment Architecture Process" an analysis of the deployment architecture process according to distributed cognition theory, where each element participating in the process is analyzed as an individual cognitive unit, and they identify challenges and difficulties that may hinder the process and compromise the quality of its outcomes. “Barbara Weber, Jakob Pinggera, Victoria Torres and Manfred Reichert” present a research design in their paper “Change Patterns for Model Creation: Investigating the Role of Nesting Depth” toward examining the impact of nesting depth on the cognitive complexity of change pattern usage when creating process models. Finally, “Jan Claes, Frederik Gailly and Geert Poels” present a research design in the paper “Cognitive Aspects of Structured Process Modeling” to investigate how structured process modeling affects the quality of the process model created. The paper also surveys some related cognitive theories that may help explain the causal relations between these two variables.

In addition to the presentations, we invited some of the presenters as well as additional guests to participate in a panel on “Information Systems Development Technologies and the Human Mind: Correspondence and Collision.”

We hope that the reader will find this selection of papers useful and be inspired by new ideas in the area of cognitive aspects of information systems engineering, and we look forward to future editions of the COGNISE workshop following the first edition this year.

June 2013

Irit Hadar
Barbara Weber

Organization

Organizing Committee

Irit Hadar	University of Haifa, Israel
Barbara Weber	University of Innsbruck, Austria

Program Committee

Daniel M. Berry	University of Waterloo, Canada
Kathrin Figl	WU Vienna, Austria
Marcela Genero	University of Castilla-La Mancha, Spain
Thomas Kohlborn	Queensland University of Technology, Australia
Joel Lanir	University of Haifa, Israel
Meira Levy	Shenkar College of Engineering and Design, Israel
Jan Mendling	WU Vienna, Austria
Daniel Moody	Neodata, Australia
Hajo Reijers	TU Eindhoven, The Netherlands

Workshop on Human-Centric Process-Aware Information Systems HC-PAIS

Human-centric issues are important to our understanding and development of information systems. In continuing the successful series of the HC-PAIS workshop, we are pleased to present the papers of the Human-Centric Information Systems (HCIS) 2013 workshop held in conjunction with CAiSE 2013, the 25th International Conference on Advanced Information Systems Engineering. The HCIS workshop series is dedicated to focusing on human-centric perspectives in the analysis, design, and use of information systems. The objective of the workshop is to extend the spectrum of the conference by providing researchers and practitioners with a platform for discussing research on human involvement and human integration in different types of IS, such as process-aware information systems (PAIS), business process management suites (BPMS) and workflow systems (WfS). The 2013 edition of the workshop attracted papers from nine different countries including Australia, Austria, Czech Republic, France, Germany, India, Luxembourg, Norway, and Spain. We received nine submissions, from which the four best rated papers were selected, yileding an acceptance rate of 44.45%. The HCIS contributions presented here address the following themes:

In "Enabling Personalized Process Schedules with Time-aware Process Views" Andreas Lanz, Jens Kolb, and Manfred Reichert present an approach to provide personalized schedules based on time-aware process schemas. The approach offers to individual users a comprehensive time-based view on business processes by visualizing them as extended Gantt diagrams.

Kimon Batoulis, Rami-Habib Eid-Sabbagh, Henrik Leopold, Mathias Weske, and Jan Mendling present in their contribution "Automatic Business Process Model Translation with BPMT" an evaluated technique to automatically translate business process models into different languages. The Business Process Model Translator (BPMT) supports the re-use of process models and allows employees, who work in subsidiaries of their multi national company, to access process models in foreign languages.

In "A Theoretical Basis for Using Virtual Worlds as a Personalized Process Visualization Approach" Hanwen Guo, Ross Brown, and Rune Rasmussen propose a theoretical analysis framework for reducing communication problems between business analysts and other process stakeholders. They show how processes of single-process performers can be captured in the virtual world and used as assistance for communication between the process stakeholders.

Hamzah Ritchi and Jan Mendling propose in "A Research Program for Studying the Impact of Process Representation on Risk Analysis" a set of hypotheses and a program outline for addressing the question of whether a process model can help users to better understand the underlying processes and consequently supports the analyst in performing risk assessments, particularly in the audit domain.

We thank the authors for their contributions, the Program Committee members for reviewing, and the CAiSE 2013 Workshop Co-chairs and the Organizing Committee for all their support.

April 2013

Sonja Kabicher-Fuchs
Jan Recker
Stefanie Rinderle-Ma

Organization

HCIS 2013 was a workshop collaboratively organized by the University of Vienna, Research Group Workflow Systems of Technology and the Queensland University of Technology, Information Systems School.

Workshop Chairs

Program Committee

Workshop on New Generation Enterprise and Business Innovation Systems NGEBIS

Innovation is one of the major drivers to enable European enterprises to compete in global markets, especially in a tough economic juncture. Yet innovation is an elusive term that is often used in an indefinite way. If we consider widely accepted definitions, we can see that they capture only part of the essence of innovation. An innovation process is different from a "usual" business process we find in an enterprise that is (supposedly) well defined in its activities, committed resources, etc. Innovation is a creative activity confronted with "wicked problems," i.e., problems difficult to solve because of incomplete, contradictory, and changing requirements.

The New Generation Enterprise and Business Innovation Systems (NGEBIS) workshop intends to address the area of information systems dedicated to business innovation that has been traditionally considered too fuzzy and ill-defined to be systematically tackled by using existing information systems and information engineering methods. We expect that the ideas discussed in the workshop will contribute to the development of methods to be used in the implementation of a new generation of information systems dedicated to business innovation, with particular attention to networked enterprises.

To this end, beyond the presentation session devoted to research papers, NGEBIS 2013 provided also the following sessions: (1) Discussion-oriented session, with demos of new tools and systems, posters, and emerging ideas; (2) WOW - Window on Online Workshop Forum, for online discussion on a selected Social Network (LinkedIn), that started 2 months before the workshop to identify the three hottest issues to be selected for the Knowledge Café held on site; (3) Knowledge Café: composed of three rotating parallel panels that discussed the three most relevant topics emerged during the WOW Forum.

This edition of NGEBIS received eight submissions, each of which was reviewed by at least two Program Committee members in order to supply the authors with helpful feedback. The committee decided to accept four contributions as regular papers and one as short paper. The workshop resulted in a multi disciplinary collection of contributions, addressing two main research lines. The first concerns organizational models for the governance and the knowledge management in innovation-oriented enterprises, represented by the papers: "Toward Innovative Model-Based Enterprise IT Outsourcing" and "Characteristics of Knowledge and Barriers Towards Innovation and Improvement in Collaborative Manufacturing Process Chains." The second concerns technical and formal methods for information processing in innovation projects, represented by the papers: "A Logic-Based Formalization of KPIs for Virtual Enterprises," "Cross-Domain Crawling for Innovation," and "Hybrid Modelling with ADOxx: Virtual Enterprise Interoperability Using Meta Models."

We would like to thank all authors for their contributions and the members of the Program Committee for their excellent work during the reviewing phase. We would also like to thank the organizers of the CAiSE 2013 conference for hosting the workshop and the BIVEE European Project that is the initiator of this venture that we expect to continue in the future.

June 2013

Michele Missikoff
Gash Bhullar
Neil Maiden
Fabrizio Smith

NGEBIS 2013 Organization

Workshop Chair

Michele Missikoff	Università Politecnica delle Marche, Italy

Program Chairs

Gash Bhullar	Technology Application Network Limited, UK
Neil Maiden	City University of London, UK

Organizing Chair

Fabrizio Smith	National Research Council - IASI "Antonio Ruberti", Italy

Program Committee

Arne Berre	SINTEF, Norway
Gash Bhullar	TANet, UK
David Chen	UB1, France
Martine Collard	University of Ag, France
Claudia Diamantini	Università Politecnica delle Marche, Italy
Asuman Dogac	SDRC, Turkey
Markus Eisenhauer	Fraunhofer, Germany
Serio Gusmeroli	TXT, Italy
Nenad Ivezic	NIST, USA
Paul Johanesson	KTH, Sweden
Bernhard Katzy	LIACS, The Netherlands
Larry Kerschberg	GMU, USA
Peter Lindgren	University of Aalborg, Denmark
Leszek Maciaszek	WUE, Poland
Neil Maiden	City University, London, UK
Boris Otto	University of St. Gallen, Switzerland
Simon Schlosser	University of St. Gallen, Switzerland
Marcus Seifert	BIBA, Germany
Konstantinos Zachos	City University, London, UK
Martin Zelm	CIMOSA, Germany

Second International Workshop on Ontologies and Conceptual Modeling Onto.Com

This volume collects articles presented at the second edition of the International Workshop on Ontologies and Conceptual Modeling (Onto.Com 2013). This workshop was organized as an activity of the Special Interest Group on Ontologies and Conceptual Modeling of the International Association of Ontologies and Applications (IAOA). It was held in the context of the 25th International Conference on Advanced Information Systems Engineering (CAISE 2013), in Valencia, Spain. Moreover, the workshop was designed with the main goal of discussing the role played by formal ontology, philosophical logics, cognitive sciences and linguistics, as well as empirical studies in the development of theoretical foundations and engineering tools for conceptual modeling.

For this edition, we had 11 submissions with contributors from Canada, France, Italy, Spain, Tunisia, UK, Australia, Sweden, Czech Republic, Greece, and Brazil. These proposals were carefully reviewed by the members of our international Program Committee. After this process, six articles (five full papers and one short paper) were chosen for presentation at the workshop.

In the paper entitled "Re-engineering Data with 4D Ontologies and Graph Databases," Sergio de Cesare, George Foy, and Chris Partridge propose an approach based on the BORO approach (a 4D foundational ontology and methodology) for interpreting raw data. Moreover, the article discusses how a model resulting from the interpretation and ontology-based transformation of these raw data can be mapped to a graph-based database architecture.

In "An Application of Philosophy in Software Modelling and Future Information Systems Development," Brian Henderson-Sellers, Cesar Gonzalez-Perez, and Greg Walkerden discuss the influences of different philosophical stances on conceptual modeling and modeling language engineering. A fundamental goal of this enterprise is to contribute to make explicit the (sometimes tacit) philosophical assumptions taken by information systems researchers and developers.

In "Knowledge Organization and the Conceptual Basis for Building Classification Systems for Complex Documents: An Application on the Brazilian Popular Song Domain," Rodrigo De Santis elaborates on the multidisciplinary and multidimensional nature of complex documents. In particular, the article discusses theoretical grounding and presents an ontology-based system for the classification of popular songs.

In "Non-monotonic Reasoning in Conceptual Modeling and Ontology Design: A Proposal," Giovanni Casini and Alessandro Mosca present a formal proposal for introducing non-monotonic reasoning in ORM (Object-Role Modeling) schemas, enriching the language with a new set of syntactic constructs.

In "Supporting Customer Choice with Semantic Similarity Search and Explanation," Anna Formica, Michele Missikoff, Elaheh Pourabbas, and Francesco Taglino present a semantic search method based on the information content approach for helping customers to make their choices. Moreover, the authors elaborate on how the method proposed could also be used to provide to the user an explanation for the ranked list of options returned from a semantic search. Finally, the paper investigates graphical representations for better representing these results.

Finally, in the paper entitled "Towards a Sociomaterial Ontology," Maria Bergholtz, Owen Eriksson, and Paul Johannesson present an ontology of sociomaterial entities, discussing how organizational entities are grounded in physical ones and how they can be understood in their approach of deontic notions such as privileges, duties, and powers. Moreover, the authors discuss impacts of these ontological notions in the practice of conceptual modeling.

We would like to thank the authors who considered Onto.Com as a forum for presentation of their high-quality work. Moreover, we thank our Program Committee members for their invaluable contribution with timely and professional reviews. Additionally, we are grateful to the support received by the IAOA (International Association for Ontologies and Applications). Finally, we would like thank the CAISE 2013 Workshop Chairs and Organizing Committee for giving us the opportunity to organize the workshop in this fruitful scientific environment.

June 2013

Giancarlo Guizzardi
Oscar Pastor
Yair Wand

Organization

Program Committee

Andreas Opdahl	University of Bergen, Norway
Brian Henderson-Sellers	University Technology Sydney, Australia
Claudio Masolo	Laboratory for Applied Ontology (ISTC-CNR), Italy
Frederick Gailly	Ghent University, Belgium
Giancarlo Guizzardi	Federal University of Espírito Santo, Brazil and ISTC-CNR, Italy
Jeffrey Parsons	Memorial University of Newfoundland, Canada
João Paulo Andrade Almeida	Federal University of Espírito Santo, Brazil
Leo Obrst	MITRE Corporation, USA
Maria Luiza Machado Campos	Federal University of Rio de Janeiro, Brazil
Nicola Guarino	Laboratory for Applied Ontology (ISTC-CNR), Italy
Oscar Pastor	Polytechnic University of Valencia, Spain
Paul Johannesson	Stockholm University, Sweden
Peter Green	University of Queensland, Australia
Pnina Soffer	University of Haifa, Israel
Ricardo Falbo	Federal University of Espírito Santo, Brazil
Richard Dapoigny	University of Savoie, France
Roel Wieringa	University of Twente, The Netherlands
Sergio de Cesare	Brunel University, UK
Vinay Chaudri	SRI International, USA
Yair Wand	University of British Columbia, Canada

First International Workshop on Variability Support in Information Systems VarIS

Contemporary off-the-shelf enterprise information systems (IS) do not provide the flexibility required by agile enterprises whose businesses need to be rapidly adapted in response to market or environmental changes. Usually, standardized IS are deployed in different companies acting in different domains and distributed over various countries not taking their inherent differences into account. Despite these differences, certain IS functionality is shared among enterprises (e.g., the invoice checking and approval process or the customer entity). While major progress has been achieved by shifting from function- to process-centric system design, currently, the construction of IS dealing with all these particularities and commonalities is far from being realized. A feasible direction to improve this situation is to address IS variability as a first class aspect during IS development.

Variability management is prevalent in a multitude of research fields including, for example, requirements engineering, software product lines, business and software process modeling, and product data management. However, a comprehensive approach dealing with variability in the context of IS engineering and IS management is still missing; e.g., it is not well understood how the varying requirements contribute to variability of the different artifacts emerging in the IS lifecycle (e.g., architectural specifications, process models, test cases, handbooks).

The VarIS workshop fills this gap by bringing together researchers and practitioners from different fields (e.g., requirements engineering, software product line engineering, business process management, software engineering, product data management) who need to deal with variability issues in IS not from an isolated point of view, but as an integrated part of a development project. The overall goal of the workshop is to look at variability issues from a wider perspective, trying to understand not only the techniques and languages that allow capturing and representing IS variability, but also IS variability including, for example, its drivers and economic implications.VarIS discusses the current state of ongoing research, industry needs, future trends, and practical experiences.

The first edition of this workshop attracted 11 international submissions. Each paper was reviewed by at least two members of the Program Committee. From these submissions, the top five were accepted as full papers and, in addition, another interesting submission was accepted as short paper for presentation at the workshop.

The accepted papers provide a good example that variability is present in many different aspects of IS, and different approaches and techniques need to be used to address it properly. For example, the paper by Murguzur, Sagardui, Intxausti, and Trujillo presents LateVa, an approach for managing BP variability along the modelling, execution, and evaluation phases of the BP lifecycle. Martinez-Ruiz, Ruiz, and Piattini deal with changes at the context level of software processes focusing on the project, and its characteristics, the organization, and the current laws. Ingles-Romero and Vicente-Chicote focus on the prototyping and verification of self-adaptive systems by means of a formal approach. Mori and Cleve present a research agenda to deal with data-intensive self-adaptive (DISA) systems where a classification framework for adaptation and key challenges for managing the complete lifecycle of DISA systems are discussed. Ponnalagu, Narendra, and Ghose propose a framework to deal with the development of reusable service-oriented applications. Finally, Ouali, Kraiem, Al-Khanjari, and Baghdadi propose a process that combines the use of maps and feature diagrams for the development of SPL.

In this first edition of the workshop, we hope that the reader will find this selection of papers useful to keep track of the latest advances and challenges in the area of variability in information systems, and we look forward to bringing new advances in future editions of the VarIS workshop.

June 2013

Victoria Torres
Félix García
Manfred Reichert

Organization

Organizing Commitee

Félix García	Universidad de Castilla-La Mancha, Spain
Manfred Reichert	University of Ulm, Germany
Victoria Torres	UniversitatPolitècnica de València, Spain

Program Committee

Danilo Caivano	University of Bari, Italy
Jaelson Castro	Universidade Federal de Pernambuco, Brazil
Remco Dijkman	Eindhoven University of Technology, The Netherlands
Marlon Dumas	University of Tartu, Estonia
Martin Eigner	TU Kaiserslautern, Germany
Joan Fons	Universitat Politècnica de València, Spain
Jon Atle Gulla	Norwegian University of Science and Technology, Norway
Alena Hallerbach	Daimler TSS, Germany
Marta Indulska	The University of Queensland, Australia
Marcello La Rosa	Queensland University of Technology, Australia
Jaejoon Lee	Lancaster University, UK
Richard Lenz	University of Nurnberg-Erlangen, Germany
Mathias Lohrmann	KPMG, Germany
Raúl Mazo	Université Paris 1 Panthéon - Sorbonne, France
Jan Mendling	WU Vienna, Austria
Jürgen Münch	University of Helsinki, Finland
Selmin Nurcan	Sorbonne Graduate Business School, France
Vicente Pelechano	Universitat Politècnica de València, Spain
Hajo Reijers	Eindhoven University of Technology, The Netherlands
Francisco Ruiz	University of Castilla-La Mancha, Spain
Antonio Ruiz-Cortés	University of Seville, Spain
Lucinéia Thom	University of Porto Alegre, Brazil
Barbara Weber	University of Innsbruck, Austria
Matthias Weske	University of Postdam, Germany

Workshop on Information System Security Engineering
WISSE

Modern information systems support significant areas of human society that require storage and processing of sensitive personal and organizational information. Therefore, developers of information systems are currently faced with important challenges related to the security of such systems. The scientific community has realized the importance of aligning information systems engineering and security engineering in order to develop more secure information systems.

The International Workshop on Information System Security Engineering (WISSE) aims to provide a forum for researchers and practitioners to present, discuss, and debate, on one hand the latest research work on methods, models, practices and tools for secure information systems engineering, and on the other hand the relevant industrial applications, recurring challenges, problems and industrial-led solutions in the area of secure information systems engineering.

This third edition of the workshop, held in Valencia (Spain) on June 18, 2013, was organized in conjunction with the 25th International Conference on Advanced Information Systems Engineering (CAiSE 2013). In order to ensure a high-quality workshop, each submitted paper went through an extensive review process. Five submissions were accepted as full papers and three as short papers. The accepted submissions address a large variety of issues related to secure information systems engineering.

We wish to thank all the contributors to WISSE 2013, in particular the authors who submitted papers and the members of the Program Committee who carefully reviewed them. We express our gratitude to the CAiSE 2013 Workshop Chairs, for their helpful support in preparing the workshop. Finally, we thank our colleagues from the Steering Committee, Nora Cuppens, Jan Jürjens, Carlos Blanco and Daniel Mellado, for initiating the workshop and contributing to its organization.

June 2013

Nadira Lammari
David G. Rosado
Haralambos Mouratidis

Organization

General Chair

Haralambos Mouratidis	University of East London, UK

Program Chair

Nadira Lammari	Conservatoire National des Arts et Métiers, France
David G. Rosado	University of Castilla-La Mancha, Spain

Steering Committee

Nora Cuppens	Telecom-Bretagne, France
Jan Jürjens	Technical University of Dortmund, Germany
Nadira Lammari	Conservatoire National des Arts et Métiers, France
Haralambos Mouratidis	University of East London, UK
David G. Rosado	University of Castilla-La Mancha, Spain
Carlos Blanco	University of Cantabria, Spain
Daniel Mellado	Spanish Tax Agency, Spain

Publicity Chairs

Shareeful Islam	University of East London, UK
Luis Enrique Sánchez	University of Castilla-La Mancha, Spain
Frédéric Gervais	Université Paris-Est Créteil, France

Program Committee

Antonio Maña	University of Malaga, Spain
Brajendra Panda	University of Arkansas, USA
Carlos Blanco	University of Cantabria, Spain
Christos Kalloniatis	University of the Aegean, Greece
Cuppens Nora	LUSSI/SERES Telecom-Bretagne, France
Daniel Mellado	Spanish Tax Agency, Spain
David G. Rosado	University of Castilla-La Mancha, Spain
Eduardo Fernández-Medina	University of Castilla-La Mancha, Spain

Eduardo B. Fernández	Florida Atlantic University, USA
Eric Dubois	CRP Henri Tudor, Luxembourg
Ernesto Damiani	Università degli Studi di Milano, Italy
Federica Paci	Università di Trento, Italy
Frédéric Cuppens	Telecom Bretagne, France
Frédéric Gervais	Université Paris-Est Créteil, France
Günther Pernul	University of Regensburg, Germany
Haralambos Mouratidis	University of East London, UK
Indrakshi Ray	Colorado State University, USA
Isabelle Comyn-Wattiau	CNAM Paris, France
Jacky Akoka	CNAM Paris, France
Jaejoon Lee	Lancaster University, UK
Jan Jürjens	Technical University of Dortmund, Germany
Javier López	University of Málaga, Spain
Kouichi Sakurai	Kyushu University, Japan
Luis Enrique Sánchez	University of Castilla-La Mancha, Spain
Marc Frappier	University of Sherbrooke, Québec
Nadira Lammari	CNAM Paris, France
Nineta Polemi	University of Pireaus, Greece
Nora Cuppens	Telecom Bretagne, France
Régine Laleau	LACL, Université Paris-Est Créteil, France
Sabrina De Capitani di Vimercati	Università degli Studi di Milano, Italy
Shareeful Islam	University of East London, UK
Stefanos Gritzalis	University of the Aegean, Greece
Steven Furnell	Plymouth University, UK
Yves Ledru	LIG, University of Grenoble, France

Additional Reviewers

Francisco Moyano	University of Malaga, Spain
Carmen Fernández-Gago	University of Malaga, Spain
Christian Broser	University of Regensburg, Germany

Table of Contents

Approaches for Enterprise Engineering Research (AppEER 2013)

International Workshop on BUSiness/IT ALignment and Interoperability (BUSITAL 2013)

International Workshop on Cognitive Aspects of Information Systems Engineering (COGNISE 2013)

Workshop on Human-Centric Information Systems (HC-IS 2013)

Next Generation Enterprise and Business Innovation Systems (NGEBIS 2013)

International Workshop on Ontologies and Conceptual Modeling (OntoCom 2013)

International Workshop on Variability Support in Information Systems (VarIS 2013)

International Workshop on Information Systems Security Engineering (WISSE 2013)

Towards Self-development of Evolutionary Information Systems: An Action Research of Business Architecture Development by Students in Socially Networked Groups

Mart Roost, Kuldar Taveter, Karin Rava, Jaak Tepandi, Gunnar Piho,
Rein Kuusik, and Enn Õunapuu

Dep. of Informatics, Tallinn Univ. of Technology, Akadeemia tee St.
15A, Tallinn 12618, Estonia
{Mart.Roost,Kuldar.Taveter,Karin.Rava,Jaak.Tepandi,
Gunnar.Piho,Rein.Kuusik,Enn.Ounapuu}@ttu.ee

Abstract. We present a case study of business architecture development by students working in socially networked groups. In this case study we emulated a self-development of an evolutionary information system. The "client system" in this emulated project was medical laboratory information system. In the role of the "change agent" were students of two different specialties: medical technology students (one group) and IT students (another group). We describe the process and results of the first (finished) phase of strategic analysis where the initial business architecture was developed. Later on this business architecture will be utilized as a platform for (social, self-) development of business processes and software. Medical technology students (knowing the problem) played the business process owner/analyst dual role. IT students (knowing IT-related solution patterns for the problem and processes) played the business designer role. The relationships between (and inside) the two groups/communities were managed using Google Sites (social) software.

Keywords: Enterprise Information Systems, Evolutionary Information Systems, Social Self-Development, Social Software, Action Research, Enterprise Architecture, Business Architecture, Methodology.

1 Introduction

The global information society has given rise to dynamic *networked organizations* (NWO) and enterprises. NWO is a term that is used to describe a variety of new emergent organizational structures such as *virtual and learning organizations* [1, 2, 3], which operate and evolve on the level of *information system* (IS). We can use the term of NWO as a synonym for a "contemporary or future organization". The success of such organizations depends on their ability to adapt to the environment and learn.

In the process of organizational learning, organizations are restructuring many relationships internally and externally to respond to the demands of a shifting market [4]. Internally companies disaggregate into smaller units focused on well-defined market opportunities. Externally companies increasingly partner with other

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 1–15, 2013.

organizations and form extended enterprises. *Extended enterprise* is a kind of NWO that forms on the level of *enterprise IS* (EIS) and embraces members from both its internal and external environments. From the above description follows that such NWO consist of relatively autonomous sub-units. Because of that, they can be adequately described by using the concept of agents. We define an agent as “an entity that performs a specific activity in an environment of which it is aware and that can respond to changes” [5]. Agents (in the same environment) can form multi-agent systems [5]. In the context of organizational learning, the agent may be termed as “change agent” and the environment as “client system” that can respond to organizational changes [6]. NWO is then a multi-agent system that forms (operates and evolves) in some IS environment as a result of the system- or development work [7] performed by agents. NWO forms on the basis of core competences and resources of the organization and is owned by independent agents. Each constituent agent of a NWO can play different roles and manages a set of business rules. Agents of a NWO form social networks [8], which can be managed with social software [9].

Social networks (in general) deal with agent relationship management. In the context of extended enterprise, social networks can be described using *business process management* (BPM) [10] lingua. We interpret a business process as a (specific) relationship/collaboration between two or more agents, playing the roles. In current case the roles are either business process owner/analyst role (medical students) or business designer role (IT students).

The main operational and learning environment for a networked organization and its business processes is its information system (IS). IS should be *evolutionary* [11]. The evolutionary information system is able to survive over time and has built-in support to handle evolutionary changes [12]. Therefore, in addition to traditional information and communication services of IS, the evolutionary IS should also provide and support services for the development of business processes and software.

Self-development of an IS is defined as a decentralized IS development model, where the whole system is not directly developed, but each agent develops its own part or role as an area of responsibilities in this whole [13, 14]. We define a *self-development subsystem* as a part of an evolutionary IS that is responsible for handling all aspects and services for the evolutionary self-development (change) of IS. Both, the evolutionary IS and its self-development subsystem, are socio-technical systems [15] with a social (business) and a technical (IT) parts. In the self-development subsystem we utilize the *End-User Development* (EUD) model in the context of networked organizations [16, 17].

Social self-development (SSD) is a community-supported self-development of a socio-technical system based on social networks. In this article we present a case study of enterprise information system self-development by students working in socially networked groups. In this case study we emulated a social self-development of an evolutionary information system. This case study has been conducted according to the *action research* (or *action learning*) methodology described in [6]. The “client system” in this emulated project was medical laboratory information system. In the role of the “change agent” were students of two different specialties: medical technology students forming one group and IT students making up another group. We describe the process and results of the first phase of strategic analysis where the initial business architecture was developed. Later on this business architecture will be utilized as a platform for social self-development of evolutionary information systems. We focus on analyzing the relationships between our social self-development and the action research methodologies (processes). The main research problem is: how do we research and develop evolutionary IS in enterprises.

In section 2 we introduce our *action research framework*. In section 3 the main results and lessons learned are presented. Section 4 provides conclusions.

2 An Action Research Framework

According to Checkland [18], any piece of research may be thought of as entailing a particular framework of ideas F that is used in a methodology M to investigate some area of interest A. In the research reported in this article, A is *evolutionary enterprise information system*, F is *social self-development*, and M is *action research*. The initial research question or problem is: how these three topics are related and work together?

2.1 Area of Interest: Evolutionary Enterprise Information Systems

Our research area (A) is *evolutionary enterprise information systems*. An evolutionary IS is able to survive over time and has built-in support to handle evolutionary changes. Information system is a *socio-technical system* that has social (business) and technical (IT) parts. An evolutionary information system can be continuously changed by its constituent agents. Agents can be human agents (persons, organizational units, groups of persons) as well as "man-made" software agents. This means that evolutionary IS can be understood and described as a multi-agent system [5].

The evolution has two "sides": continuous work improvement [19] and software evolution [20]. Business process evolution, which is a sub-domain of the *business process management* (BPM) [10] domain, bridges both evolution sides [21]. IS evolution is often considered as being only (or mainly) the evolution of the software [22]. This kind of software evolution is often considered at the later stages of software development (implementation, execution), mostly adopting pragmatic approaches [22, 23] only. Our approach is an architectural approach.

Our starting point is that evolution is a matter of architecture. System's architecture consists "fundamental concepts or properties of a system in its environment embodied in its elements, their relationships, and in the principles of its design and evolution" [24]. Evolution of enterprise IS is a matter of *enterprise architecture* (EA). Enterprise architecture is the "[enterprise] conceptualization of the form, function, and fitness-for-purpose of a system in its environment, as embodied in the elements of the system, the relationships between those elements, the relationship of the system to its environment and the principles guiding the design and evolution of the system" [25].

In *model driven development* (MDD) approaches [26, 27], an EA is described by a meaningful and useful set of enterprise models [25, 28] for (and owned/evolved by) large sub-communities of the enterprise. The evolution of the models is guided by a meta-model that describes a more stable but in principle evolving meta-architecture

[28]. If this meta-architecture is implemented as a core subsystem of EIS, we have *evolving EA* [29].

The business architecture [30] is a part of an EA, which's describing set of models corresponds to the first two rows of the Zachman Framework [31, 32], representing respectively the Executive Perspective and Business Management Perspective.

To handle model-driven evolution, an evolutionary IS should also provide and support services for the development of business processes and software, in addition to traditional information and communication services If the development is performed by agents of an IS in a social manner, we can talk about social self-development, which is addressed by next subsection.

2.2 Framework of Ideas: Social Self-development

Our framework of ideas (F, Checkland [18]) is *social self-development* that combines ideas from *end-user development* [33], *developmental work research* [19], *meta-design* [34], and *infrastructuring* [35]. Infrastructuring is defined as a bottom-up, participatory approach to EA development.

The solution of a *self-development subsystem* that was defined in section 1 can be based on the meta-design (MD) theory. According to MD, one of the main weaknesses of designing (architecting) is related to evolutionary character of the design (architecture) and the incapacity of fully anticipating at design-time the needs and tasks of users [36].

With the purpose of overcoming this limitation, MD aims at defining mechanisms that allow "owners of the problem" to become designers [37]. MD is defined as a conceptual framework that allows end-users to create contents by using socio-technological infrastructure, in which people can actively participate [38]. We define enterprise IS self-development subsystem as an enterprise-wide MD framework, which follows the concepts of *evolving enterprise architecture* and *infrastructuring*.

The MD has become a leading theory for end-user development (EUD) [34, 35]. The EUD is defined in [33] as "the set of methods, techniques, and tools that allow users of software systems, who are acting as non-professional software developers, at some point to create or modify a software artifact". In the context of enterprise IS that we interpret as socio-technical systems, the EUD requires a complementary method for continuous work improvement. Syrjanen and Kuutti have proposed *developmental work research* [19] as a potential method to be coupled with EUD to deal with work improvement aspects needed to implement EUD in organizational environments.

Our architectural approach to *social self-development* (SSD) of *evolutionary information systems* requires proper methods and work products for continuous work improvement supported by EUD of software applications.

We see social self-development defined in section 1 as a method for both EUD and MD that is applicable in the context of networked organizations. An important goal of our research is to apply and extend MD theory in the context of extended enterprises and evolutionary information systems in order to describe and implement a platform for SSD of EIS.

2.3 Methodology: Action Research

Our research is conducted according to the Action Research methodology [39]. Action Research (AR) is known by many other names like participatory research, collaborative inquiry, emancipatory research, action learning, and contextual action research. AR can be seen as a kind of practical problem solving approach. The essence of the approach is “learning by doing” – a group of people identify a problem, do something to resolve it, see how successful their efforts were, and if not satisfied, try again [39]. AR aims to contribute both to the practical concerns of people in an immediate problematic situation and to further the goals of (social) science simultaneously [40]. According to Reason & Bradbury in [41], AR is an interactive inquiry process that balances problem solving actions implemented in a collaborative context with data-driven collaborative analysis or research to understand underlying causes enabling future predictions about personal and organizational change).

AR occurs as collaboration between a “client system” and a “change agent” (who is opposed to traditional “observer”). For AR performed by our research group, a “client system” is a concrete enterprise for which we perform problem domain analysis and the “change agent” is our university’s research group. It is important that in our SSD approach, not only an external research group, but each agent in the context of an (extended) enterprise is seen as a change agent who performs the needed changes in the context of its own roles and responsibilities defined by them. This is in harmony with the Developmental Work Research approach [19], which we also view as a special version of AR.

2.4 Case Study: Action Learning on Medical Laboratory

As an example case study for our AR, this article presents a real action learning process in the context of the subject ”Teamwork in Information Systems' Development” taught at our university. In the case study conducted, the ‘client system’ was Medical Laboratory with its Laboratory Information Management System (LIMS). In the role of the ‘change agent’ were students of two different specialties: students of medical technology forming one group and IT students making up another group. In this article we describe the process and work products of the first finished phase of problem domain analysis that produced initial business architecture. Within this process, the students of medical technology played the dual roles of core business process owners and business analysts knowledgeable about the problem. The IT students played the roles of business designers knowledgeable on possible IT-enabled solutions for the problem.

The relationships between and within the two groups and also work products created and changes introduced by them were managed by Google Sites social software. In the context of each agent’s role, which was defined in terms of its responsibilities, a diary was written, which reflected the process of playing and developing this role (analogously with the situation in a theatre, where each actor would write a web-based diary or blog between performances; an entry in the diary would usually reflect (from the viewpoint of the writer) a particular performance (but sometimes might express more general knowledge). Such learning model is in good harmony with the Action Research methodology.

2.5 Limitations

An important limitation of the work is that we simulated the action research setup with students – thus it would be rather a kind of lab experiment because students “played” practitioners. That is close to but not the same like “really doing it”. Therefore we plan to repeat similar action research with a real enterprise.

3 Social Self-development within Enterprise Business Architecture: Our Methodology

In this section we present the main results of and lessons learned from the *action research* project introduced in the previous section. The results include the work products of the problem domain analysis of a medical laboratory and its IS (LIMS).

At the meta-level, our research results also include the problem domain analysis of our problem domain analysis methodology and its IS (as a part of LIMS Self-Development Subsystem, which includes a software factory described in [42]).

3.1 Strategic Analysis of the EIS (LIMS)

In this section we describe the process and work products of the first finished phase of the problem domain analysis that produced an initial version of the business architecture descriptions of the enterprise - Medical Laboratory. We will next view our methodology and its usage from three perspectives: Product Perspective, Process Perspective, and Self-Development Perspective.

The Product Perspective. According to our problem domain analysis methodology [43, 13], the overall work product termed as Enterprise Architecture (EA) is composed of the following three sub-architectures:

- Business Architecture;
- Technology Architecture;
- Development (Work) Architecture.

In the context of the student project, we focused on describing the Business Architecture. A snapshot of the description of the Business Architecture of the Medical Laboratory is depicted in Figure 1.

According to our methodology, a Business Architecture is described by the following three interrelated views:

- Organizational view;
- Functional view;
- Informational view.

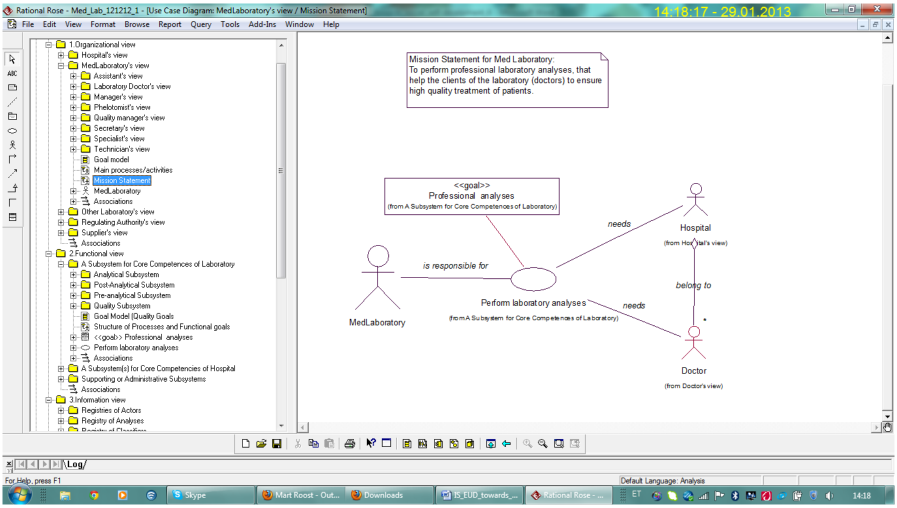


Fig. 1. An overview of the Business Architecture of the Medical Laboratory as described in the student project

The Organizational view reflects the organizational decomposition of a business, and consists of Organizational Subsystems. An Organizational Subsystem of EIS describes an Agent's Role in terms of responsibilities of an Agent playing the corresponding role in the context of the EIS as a socio-technical system. In the context of the Medical Laboratory, examples of Organizational Subsystems are Hospital Department, Laboratory Specialist, and Regulative Authority.

An Organizational Subsystem can be represented by a software agent [5], for example, by a Laboratory Specialist's Agent, which assists the respective human agent in fulfilling its professional responsibilities as well as the responsibilities that are concerned with the (self-)development of the EIS. An example of an Organizational Subsystem's description is given in Figure 2.

The Functional view reflects the process decomposition of an enterprise, and consists of Functional Subsystems. A Functional Subsystem of an EIS describes and implements a major Business Process as a potential service in the context of the EIS as a socio-technical system. In the context of the Medical Laboratory, examples of Functional Subsystems are Pre-Analytical Subsystem, Analytical Subsystem, and Quality Subsystem, which describe and implement the respective Pre-Analytical, Analytical, and Quality Management Business Processes. According to the theory of Meta-Design proposed in [34], such Functional Subsystems can be handled as Business Process Archetypes.

The Informational view reflects the structure of conceptual objects (informational architecture) of an enterprise, and consists of Informational Subsystems called Registries of EIS. A Registry describes and/or implements a major Business Object in the context of an EIS as a socio-technical system. In the context of the Medical Laboratory, examples of Informational Subsystems are Registry of Samples, Registry of Analyses, and Registry of Equipment. Such Registries are similar to Business Archetype Patterns proposed in [42].

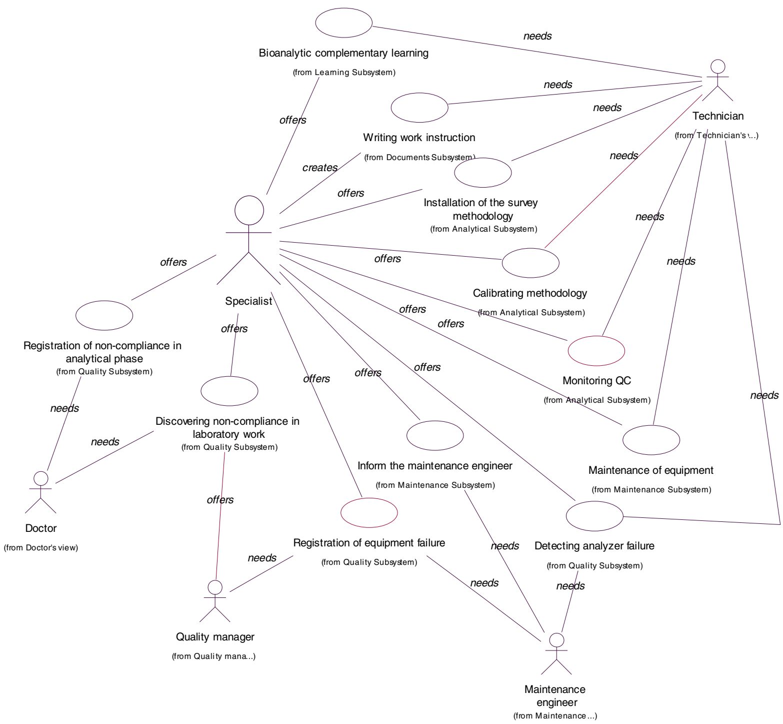


Fig. 2. An overview of an Organizational Subsystem of the Medical Laboratory described in the student project: the responsibilities of Laboratory Specialist and its relations to other roles

The Process Perspective. We next address the Process Perspective of Enterprise Business Architecture Development.

According to our methodology, the Enterprise Business Architecture Development process is owned by a player of the Enterprise Business Architect role, which is responsible for the overall Business Architecture of the Enterprise.

The Enterprise Business Architect works together with players of the Business Analyst and Business Designer roles.

A Business Analyst is responsible for models and descriptions of one or more Organizational Subsystems of an enterprise.

A Business Designer is responsible for models and descriptions of one or more Informational Subsystem(s) (Registries).

We decompose the process of Enterprise Business Architecture Development proposed by us (v. Figure 3) into the following three sub-processes:

- Strategic Business Analysis (performed by all Business Analysts)
- Strategic Business Design (performed by all Business Designers)
- Managing and Evolving the Whole (performed by the Business Architect).

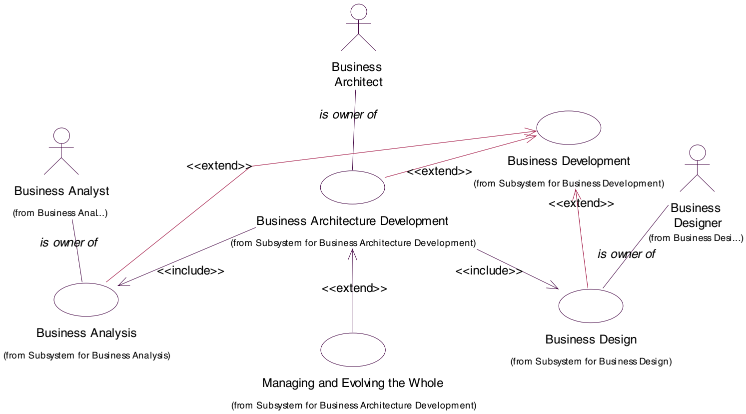


Fig. 3. An overview of Enterprise Business Architecture Development

Strategic Business Analysis is a part of the more general process of Business Analysis that is performed in the problem domain analysis phase of EIS development. Strategic Business Analysis is performed in the context of concrete Organizational Subsystems to be analyzed that are assigned to concrete Business Analysts. In the perspective of Self-Development, our goal is to educate and support business actors (agents) who are able and interested in acting as Business Analysts of their own Organizational Subsystems. For example, a Laboratory Specialist in a Medical Laboratory represented in Figure 2 should be able and interested in analyzing and modeling the Organizational Subsystem in the problem sub-domain of laboratory work *in concert with the responsibilities defined by his/her role*. In our student project, the students of medical technology played the dual roles that combine a laboratory-specific role with the Business Analyst's role. Each student was responsible for one or two Organizational Subsystems. Strategic Business Analysis of an Organizational Subsystem was divided into the following two major phases in a "bottom-up" manner:

- In the first phase of strategic business analysis, a student "playing" a laboratory-specific role described and modeled the vision of his/her Organizational Subsystem individually (in principle, such a part might alternatively be played and described by a team or community);
- In the second phase of strategic business analysis, the student coordinated her/his vision/descriptions with "players" of related laboratory-specific roles (this is a part of the "horizontal coordination" activity over the whole Organizational view), and established the necessary change requests in order to change concrete Functional and Informational Subsystems. For example, the student who played the Laboratory Specialist role, coordinated her work products (one of them is depicted in Figure 2) with the players of the roles of Technician, Maintenance Engineer, Quality Manager, and Doctor; and created requirements and change requests for the IT students, who served as Business Designers of the Quality, Maintenance, Analytical, Learning, and Documents Subsystems (v. Figure 2). During the second

phase, sub-communities of cooperating Business Analysts and Business Designers were formed around the Organizational Subsystems, with the help of the social software.

In the context of the Action Research (AR) methodology, which was briefly described in sub-section 2.4, the “bottom-up” sub-process of Strategic Business Analysis corresponds to the collaborative analysis and research activity with the purpose to understand one’s area of concern or problem sub-domain. For the latter, the “top-down” sub-process of Business Design creates an IT-enabled solution. Strategic Business Design is the part of the more general process of Business Design that is performed in the context of the Strategic Analysis phase of EIS development. Strategic Business Design is focused on major business objects that are assigned to concrete Business Designers. The business objects are primarily described as components of Informational Subsystems or Registries. The lifecycles of business objects are often managed by Functional Subsystems. For example, a central business object of the Laboratory is the biochemical Analysis (test), which is described by the Registry of Analyses and managed by the Analytical Subsystem. A Business Designer, who is responsible for specific business objects and Functional Subsystems related to them, can apply domain-specific or even more general solution patterns, which can be adapted differently in the contexts of different Organizational Subsystems, and even in different organizations (Laboratories). In the perspective of Self-Development, Business Designers belong to the context of the extended enterprise (Laboratory) as providers of respective development services supporting Social Self-Development.

In our student project, the role of Business Designer was played by IT students. Each IT student was responsible for two or three business objects and their respective Registries and Functional Subsystems. For example, one IT student was responsible for the Registry of Equipment, the Registry of Maintenance Works, and for the Maintenance Subsystem. This student worked in pair with a student of medical technology who represented the (Manager of the) Maintenance Department. The Business Design for these business objects and subsystems was also divided into two major phases, but in the following “top-down” manner:

- In the first phase of strategic business design, some general solution pattern for the business object and its Functional Subsystems was introduced by the Business Designer, and was integrated into the whole business architecture in collaboration with the Business Architect (played by one of the authors of the paper), and with the Business Designers of related objects and Functional Subsystems;
- In the second phase of strategic business design, which occurs in the context of concrete Organizational Subsystems, the respective elements of the Organizational Subsystems were described on the basis of the respective elements of the Registries and Functional Subsystems that describe and implement the business object. For example, the Laboratory Specialist’s view to his/her Organizational Subsystem was described on the basis of the elements of the Analytical, Quality, Documents, and Maintenance Subsystems, as is shown in Figure 2. Good collaboration within pairs of a Business Analyst and a Business Designer (for example, the Laboratory

Specialist and the Business Designer of the Documents), supported by the social software was regarded as a key to success of this phase.

The Managing and Evolving the Whole is the part of the Enterprise Business Architecture Development that is performed by the Enterprise Business Architect. This includes the tasks of building an initial structure of the architectural description that was introduced above as the Product Perspective of our methodology, deciding the division of labor according to Functional Subsystems between individual Business Analysts and Business Designers, supporting and coordinating the collaboration between individual Business Analysts and Business Designers, monitoring and changing the structure of the whole Enterprise Business Architecture, etc.

The Self-development Perspective. In the Self-Development perspective of the Strategic Analysis Methodology, we consider the methodology proposed by us as a part of the Meta-Design (MD) framework that includes the domain-independent Meta-Design layer, the domain-specific Design layer, and the agent-specific Usage layer. These layers form a separate “abstraction-concretization” dimension that is orthogonal to the Organizational, Functional, and Informational views of the Enterprise Business Architecture described above. We plan to implement this MD framework as the Self-Development Subsystem of an EIS supporting the Social Self-Development (SSD) of Business Processes and their underlying software applications in a similar fashion, but with a much higher quality compared to the student project described in this article. In the student project, the supporting infrastructure (software) included only the UML modeling software for describing and managing the business architecture, and the social software for relationship and change management in the form of blogs. The future Self-Development Subsystem will consist of subsystems for the following purposes: for Business Development (including Business Architecture Development, Strategic Business Analysis, and Strategic Business Design), for Technology Development (including Software Factories described in [42]), for Evolutionary Change Management (including the blogs similar to the ones used in the student project), and for Simulation and Training.

3.2 Social Self-Development (of Enterprise Business Architecture) as Action Research

In our methodology, the EIS evolution is handled according to the general learning model of Action Research, which was described in sub-section 2.3, as follows:

- Agents of an (extended) enterprise, who play the dual roles of some business concepts’ owners (v. the Product Perspective in section 3.1) and developers (Business Analyst, Business Designer, Business Architect), serve as ‘change agents’;
- The enterprise with its IS serves as the ‘client system’;
- The ‘bottom-up’ sub-process of Business Analysis serves as the ‘data-driven collaborative analysis or research’ (v. section 2.3);
- The ‘top-down’ sub-process of Business Design serves as the ‘collaborative problem solving or change’, in the context of AR (v. section 2.3).

The version of AR proposed by us follows the SSD approach and its meta-model, both of which have been introduced in [17].

3.3 Lessons Learned from the Action Research

The following major lessons were learned from the student project described in the current section:

- The approach (SSD) is applicable in the collaborative learning contexts that are similar to our student project, but requires much richer supporting infrastructure;
- Using the social software based on blogs is very useful in such contexts;
- Similarly to the pair programming, we can also perform strategic analysis and design effectively in pairs of a “business person” (like a student of medical technology) and an “IT-person” (like an IT student);
- Without continuous community-based modeling activity the SSD approach does not work.

4 Conclusion and Future Work

The research problem was: how do we research and develop evolutionary information systems of enterprises. In this paper, an approach to social self-development of evolutionary information systems was described. This approach was described by relying on the example of a real action research project that was performed in the context of the subject ‘Teamwork in information systems development” taught at our university. The ‘client system’ was Medical Laboratory with its Laboratory Information Management System (LIMS). In the role of the ‘change agent’ were students of two different specialties: students of medical technology, forming one group and IT students, making up another group. The students of medical technology played the dual roles of core business process owners and business analysts, who are knowledgeable about the problem. The IT students played the roles of business designers knowledgeable on IT-enabled possible solutions for the problem. The relationships between and within the two groups or communities were managed by using Google Sites social software.

We described the process and work products of the first phase of problem domain analysis that produced descriptions and work products of the business architecture of the enterprise under discussion – Medical Laboratory. We interpret the enterprise business architecture development methodology proposed by us as a core part of the resulting business architecture. Such architecture follows the concepts of meta-design, evolving enterprise architecture, and infrastructuring, and serves as a platform for further social self-development of business processes of an enterprise and their underlying software applications. The social self-development was defined as an agent-centric and community-supported development of a socio-technical (multi-agent) system by means of social networks that are based on social software.

In the near future, we will design and develop a solution to self-development, where software agents representing organizational units or even individual roles engaged in self-development manage and coordinate the necessary collaboration and competition between them. Principles for this kind of solution are, for example, described in [44]. We will also elaborate the model-based description of our social self-development methodology, design its supporting toolset, and plan to use them in real enterprises engineering situations. Our methodology can be understood as a domain-specific version of the action research methodology, which is applicable for evolutionary information systems.

References

1. Kimble, C.: Management and Information Systems (March 30, 2007), `http://www.chris.kimble.com/Courses/mis` (retrieved October 4, 2009)
2. Strausak, N.: Resumee of VoTalk, Organizational Virtualness. In: Sieber, P., Griese, J. (eds.) Proceedings of the VoNet-Workshop, April 27-28, Simowa Verlag Bern (1998), `http://virtual-organization.net/news/proc-98.pdf` (retrieved May 11, 2001)
3. Senge, P.M.: A Guide to Manage Knowledge: The Fifth Discipline: The Art & Practice of the Learning Organization. Currency-Doubleday, New York (2006)
4. Wenger, E., et al.: A Guide to Manage Knowledge: Cultivating Communities of Practice. Harvard Business School Press, Boston (2002)
5. Sterling, L., Taveter, K.: The art of agent-oriented modeling. The MIT Press, Cambridge (2009)
6. O'Brien, R.: An Overview of the Methodological Approach of Action Research (1998), `http://www.web.ca/robrien/papers/arfinal.html` (retrieved July 9, 2012)
7. Rava, K., Mikli, T.: System Work in System Development. In: Proceedings of the IFORS Special Conference (SPC8) on Organizational Structures, Management, Simulation of Business Sectors and Systems, pp. 213–217. Technologija, Kaunas (1998)
8. Scott, W.R., Davis, G.F.: Networks in and around Organizations: Organizations and Organizing. Pearson Prentice Hall, New Jersey (2003)
9. Wikipedia Foundation - Wikipedia, Social software, `http://en.wikipedia.org/wiki/Social_software` (retrieved July 9, 2012)
10. Chang, J.F.: Business Process Management Systems: Strategy and Implementation. Auerbach Publications, New York (2006)
11. Patel, N.V.: Adaptive Evolutionary Information Systems. Idea Group Publishing, London (2003)
12. Bellahsène, Z., Léonard, M. (eds.): CAiSE 2008. LNCS, vol. 5074. Springer, Heidelberg (2008)
13. Roost, M., Kuusik, R., Veskioja, T.: A Role-Based Framework for Information Systems Self-Development. In: Russo, N.L., Fitzgerald, B., DeGross, J.I. (eds.) IFIP TC8 WG8.2 International Working Conference on Realigning Research and Practice in Information Systems Development: The Social and Organizational Perspective, pp. 95–105. Kluwer Academic Publishers, Norwell (2001)
14. Roost, M., Rava, K., Veskioja, T.: Supporting Self-Development in Service Oriented Information Systems. In: Le, M.H., Demiralp, M., Mladenov, V., Bojkovic, Z. (eds.) 7th WSEAS International Conference on Applied Informatics & Communications (AIC 2007), pp. 52–57. WSEAS Press, Athens (2007)

15. Ropohl, G.: Philosophy of socio-technical systems. Society of Philosophy and Technology 4(3) (Spring 1999), `http://scholar.lib.vt.edu/ejournals/SPT/v4_n3html/ROPOHL.html` (retrieved May 11, 2012)
16. Mehandjiev, N., De Angeli, A., Wajid, U., Namoun, A., Battocchi, A.: Empowering End-Users to Develop Service-Based Applications. In: Piccinno, A. (ed.) IS-EUD 2011. LNCS, vol. 6654, pp. 413–418. Springer, Heidelberg (2011)
17. Roost, M., Piho, G., Taveter, K.: Information Systems' Self-development as a Model of End-User Development in Networked Organizations. In: Proceedings of the Second International Workshop on EUD4Services Methodologies, Tools and Evaluations, EUD4Services (June 6, 2011), `http://130.88.6.246/EUD4Services/Workshop2011/` (retrieved June 5, 2011)
18. Checkland, P., Holmes, S.: Information, Systems and Information Systems. John Wiley & Sons Ltd., London (1998)
19. Syrjänen, A.-L., Kuutti, K.: From System Development Toward Work Improvement: Developmental Work Research as a Potential Partner Method for EUD. In: Piccinno, A. (ed.) IS-EUD 2011. LNCS, vol. 6654, pp. 123–138. Springer, Heidelberg (2011)
20. Mens, T., Wermelinger, M., Ducasse, S., Demeyer, S., Hirschfeld, R.: Challenges in software evolution. In: Proceedings of 8th International Workshop on Principles of Software Evolution. IEEE Press (2005)
21. Netjes, M., Mansar, S.L., Reijers, H.A., van der Aalst, W.M.P.: An Evolutionary Approach for Business Process Redesign – Towards an Intelligent System. In: Proceedings of the Ninth International Conference on Enterprise Information Systems, ICEIS 2007, Funchal, Madeira, Portugal, June 12-16, vol. EIS, pp. 47–54 (2007)
22. Cimpan, S., Verjus, H., Alloui, I.: Dynamic Architecture Based Evolution of Enterprise Information Systems. In: Proceedings of the Ninth International Conference on Enterprise Information Systems, ICEIS 2007, Funchal, Madeira, Portugal, June 12-16, vol. EIS, pp. 221–229 (2007)
23. Demeyer, S., Ducasse, M., Nierstrasz, O.: Object-Oriented Reengineering Patterns. Morgan Kaufmann (2002)
24. International Organization for Standardization: Systems and Software Engineering – Architecture Description (ISO/IEC CD1 42010), Draft version of January 2010, currently titled Recommended Practice for Architecture Description of Software Intensive Systems, `http://www.iso.ch` (retrieved December 30, 2012)
25. International Organization for Standardization: Automation systems and integration - Framework for enterprise architectures and models (ISO WD 15704), draft version of January 2009, currently titled Industrial automation systems – Requirements for enterprise architecture methodologies (2009)
26. Krogstie, J.: Integrating Enterprise and IS Development using a Model Driven Approach. In: Vasilecas, O., et al. (eds.) Information Systems Development: Advances in Theory, Practice and Education (13th International Conference on Information Systems Development, ISD 2004), pp. 43–54. Springer, New York (2005)
27. Frankel, D.S.: Model Driven Architecture: Applying MDA to Enterprise Computing. Wiley Publishing, Indianapolis (2003)
28. Frank, U.: Enterprise Modeling (2009), `http://www.wi-inf.uni-due.de/FGFrank/index.php?lang=en&&groupId=1&&contentType=ResearchInterest&&topicId=14` (retrieved June 26, 2012)

29. Martin, R., Purao, S., Robertson, E.: Evolving Enterprise Architectures (2012), http://www.tinwisle.com/modeling/Martin_Purao_Robertson_preICEIMT_paper.pdf (retrieved June 18, 2012)
30. The Open Group, TOGAF Standard, Version 9.1 (2009), http://www.opengroup.org/togaf/ (retrieved June 28, 2009)
31. Zachman, J.A.: A Framework for Information Systems Architecture. IBM Systems Journal 26(3) (1987)
32. Zachman International – The Official Home of The Zachman Framework, The Zachman Framework for Enterprise Architecture: The Enterprise Ontology, Version 3.0 (2011), http://www.zachman.com/ (retrieved June 26, 2012)
33. Lieberman, H., Paterno, F., Wulf, V. (eds.): End-User Development. Kluwer Academic Publishers, Dordrecht (2006)
34. Fischer, G., Giaccardi, E.: Meta-Design: A Framework for the Future of End-User Development. In: Lieberman, H., et al. (eds.) End-User Development, Human-Computer Interaction Series, pp. 427–457. Springer, Berlin (2006)
35. Bolmsten, J., Dittrich, Y.: Infrastructuring When You Don't – End-User Development and Organizational Infrastructure. In: Piccinno, A. (ed.) IS-EUD 2011. LNCS, vol. 6654, pp. 139–154. Springer, Heidelberg (2011)
36. Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A.G., Mehandjiev, N.: Meta-Design: A Manifesto for End-User Development. Communications of the ACM 47(9), 33–37 (2004)
37. Costabile, M.F., Fogli, D., Lanzilotti, R., Marcante, A., Mussio, P., Parasiliti Provenza, L., Piccinno, A.: Meta-design to face co-evolution and communication gap between users and designers. In: Stephanidis, C. (ed.) HCI 2007. LNCS, vol. 4554, pp. 46–55. Springer, Heidelberg (2007)
38. Díez, D., Díaz, P., Aedo, I.: Meta-design Blueprints: Principles and Guidelines for Co-design in Virtual Environments. In: Piccinno, A. (ed.) IS-EUD 2011. LNCS, vol. 6654, pp. 276–281. Springer, Heidelberg (2011)
39. O'Brien, R.: An Overview of the Methodological Approach of Action Research (1998), http://www.web.ca/robrien/papers/arfinal.html (retrieved July 9, 2012)
40. Gilmore, T., Krantz, J., Ramirez, R.: Action Based Modes of Inquiry and the Host-Researcher Relationship. Consultation 5(3), 161 (1986)
41. Reason, P., Bradbury, H.: The SAGE Handbook. Participative Inquiry and Practice, 1st edn. Sage, London (2001)
42. Piho, G., et al.: From Archetypes Based Domain Model via Requirements to Software: Exemplified by LIMS Software Factory. In: Proceedings of the 34th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO, Opatia, Horvatia, May 23-27 (2011)
43. Roost, M., Kuusik, R., Rava, K., Veskioja, T.: Enterprise Information System Strategic Analysis and Development: Forming Information System Development Space in an Enterprise. In: Okatan, A. (ed.) Proceedings of the International Conference on Computing Intelligence, ICCI 2004, Istanbul, Turkey, pp. 215–219 (2004)
44. Huhns, M.N.: From DPS to MAS to ...: Continuing the Trends. In: Decker, Sichman, Sierra, Castelfranchi (eds.) Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems, AAMAS 2009, May 10-15. International Foundation for Autonomous Agents and Multiagent Systems, Budapest (2009)

Research Methodology for Enterprise Interoperability Architecture Approach

Wided Guédria[1], Khaled Gaaloul[1], Hend*erik* A. Proper[1,2], and Yannick Naudet[1]

[1] Centre de Recherche Public Henri Tudor, Luxembourg
[2] Radboud University Nijmegen, Nijmegen, The Netherlands
{wided.guedria,khaled.gaaloul,erik.proper,yannick.naudet}@tudor.lu

Abstract. As technology becomes more far-reaching and interconnected, the need of interoperability is becoming increasingly important. The Ontology of Enterprise Interoperability (OoEI) was defined as a scientific reference model regarding interoperability leading to a common understanding involving this topic. The OoEI was proposed in the general context of Enterprise Interoperability with high level concepts using the system theory. This needs to be enriched with concepts from Enterprise domain. The discipline of Enterprise Architecture (EA) advocates the use of models to support business services on enterprises. Among them, this work focuses on ArchiMate. In order to provide business services support, ArchiMate, should be amenable to analyze of various properties, as e.g. the interoperability requirements. This paper proposes a set of concepts covering the EA and interoperability domains. Through literature review and framework research, we identify key aspects of interoperability and EA and their associations, resulting in a reference conceptual model for integrated Enterprise Architecture Interoperability. The proposed model is defined based on the Design Science Research methodology. A case study illustration will be used for the evaluation as part of the research approach.

Keywords: Research methodology, Enterprise Interoperability, Enterprise architecture.

1 Introduction

The obligation to become more competitive and effective in providing better products and services requires enterprises to interoperate and evolve into network. *The availability of the Internet and information technologies has encouraged new business strategies that take advantage of enterprises ability to create networks or to network with other enterprises* [1]. One of the challenges faced by a network of enterprises is the development of interoperability between its members, which is highly correlated to the ability to use networked architectures to collaborate efficiently [2, 3].

Enterprise Architecture (EA) is generally considered to provide a good steering instrument to analyze the current state of the enterprise (As-is), identify and

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 16–29, 2013.

describe alternative future states (To-be), guard the cohesion and alignment between the different aspects of an enterprise such as business processes and their ICT (Information and Communications Technology) support [4]. The analysis of the As-is situation is very important to make conscious decisions about a future path design.

In this paper we propose to follow a scientific research methodology to analyze the as-is situation of an enterprise in terms of interoperability requirements from an architectural point of view. But first of all, we need to know what interoperability is. The most commonly acknowledged definition of interoperability is the one provided by 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* [5]. However, interoperability is not easy to understand due to its numerous definitions and interpretations. Ford *et al.* point out that according to their survey, thirty-four definitions of interoperability have been proposed since 1977 [6].

To deal with this problem the Ontology of Interoperability (OoI) was proposed to define the Interoperability domain [7]. It was thereafter extended by the Ontology of Enterprise Interoperability (OoEI). This OoEI aims at formally defining Enterprise Interoperability (EI) while providing a framework to describe problems and related solutions pertaining to the interoperability domain. It was proposed in the general context of EI with high level concepts using the system theory [8]. This needs to be enriched with enterprise concepts from EA domain. Moreover, EA models do not take into account interoperability concepts despite the importance of the interoperability in the survival cycle of an enterprise. In this paper we propose to integrate the two domain concepts (EA and EI) from a business point of view and to use the conceptual modeling to define the integrated domain. It is widely accepted that conceptual models are a prerequisite for successfully planning and designing complex systems [9]. Over the last decades, conceptual modeling has been employed to facilitate, systematize, and aid the process of information system engineering [10].

Based on ArchiMate as a modeling language of EA [11] and OoEI [12], a conceptual model is proposed. *A conceptual model is a typically graphical representation, hence can provide limited vocabulary* [13], constructed by IS professionals of some-ones or some groups perception of a real-world domain [14]. Conceptual modeling may be used to ease the implementation of an information system or to provide a common understanding between the organizations needs and an enterprise application. It is also *suitable to systematize knowledge, provide guiding research and map a portion of reality* [14].

The proposed model is defined based on the design-science research (DSR) methodology proposed by Hevner *et al.* [15] and Winter [16]. *DSR addresses important unsolved problems in unique or innovative ways or solved problems in more effective or efficient ways comparing to traditional or so-called routine design* [15]. Although it may seem impossible to find general and meaningful concepts for the entire domain of integrated Enterprise Architecture Interoperability, it is better to adopt the so-called constructive research strategy [17].

2 Research Methodology

This research is based on a simplification of the design-science research (DSR) as proposed by Hevner et al. [15] and Winter [16]. The methodology applied is divided according to the two processes of design science research in information system, *Build* and *Evaluate* [17]. The build process is composed by two stages whereas and the evaluation process is composed by only one stage (see Table 1).

Table 1. Research methodology

Build		**Evaluate**
Conceptual definition	Conceptual model construction	Evaluation
- Domain definition - Concepts identification - Concept definition	- Analysis of relations between concepts - Integration of the two domains concepts into one model	Use case

The first stage, conceptual definition, has two main milestones: concepts domain and domain definitions within the set up boundaries established between EI and EA. In this stage we proceed with literature study on interoperability models and frameworks together with EA modeling languages. Also, at this stage, we identify and define the concepts that we present in section 3. An analysis of the relations between concepts is required to understand the integrated model that is constructed in section 4.

The second stage, evaluation, is done based on the observational case study as described in [15]. The evaluation part is illustrated by a case study in section 5.

3 Conceptual Definition

3.1 Enterprise Interoperability

In order to understand the enterprise interoperability domain, we need to study the operational entities where interoperations take place within an enterprise. This aspect is mainly defined through various existing interoperability frameworks and models, which are reviewed as follows:

Framework for Enterprise Interoperability. *The main purpose of an interoperability framework is to provide an organizing mechanism so that concepts, problems and knowledge on enterprise interoperability can be represented in a more structured way* [18].

So far, the most known EI frameworks are: ATHENA (Advanced Technologies for interoperability Heterogeneous Enterprise Networks and Applications) Interoperability Framework (AIF) [19], the European Interoperability Framework [20] and the Framework for Enterprise Interoperability (FEI) [21].

Compared to other interoperability frameworks, the FEI provides three explicitly defined interoperability dimensions (interoperability barriers, interoperability concerns and interoperability approaches) to allow defining interoperability research domain [12]. The FEI was developed within the frame of INTEROP Network of Excellence [18] and is now published as an international standard (ISO 11354 - 1). It defines a classification scheme for interoperability knowledge according to three dimensions:

1. Interoperability Barriers: According to FEI, the establishment of interoperability consists in removing all the identified barriers. Three kinds of barriers are identified: Conceptual (syntactic and semantic differences of information to be ex-changed), Technological (incompatibility of information technologies: architecture and platforms, infrastructure, etc.), and Organizational (definition of responsibilities and authorities).
2. Interoperability Concerns: They represent the areas concerned by interoperability in an enterprise. Four concerns are defined, namely business interoperability (work in a harmonized way to share and develop business between companies despite the difference of methods, decision making, culture of enterprises, etc.), process interoperability (make various processes work together. In the inter-worked enterprise, the aim will be to connect internal processes of two companies to create a common process), service interoperability (making work together various services or applications by solving the syntactic and semantic differences) and data interoperability (make work together different data models with different query languages to share information coming from heterogeneous systems).
3. Interoperability Approaches: there are three basic ways to relate entities together to establish interoperations: The integrated approach (characterized by the existence of a common format for all the constituents systems), the unified approach, characterized by the existence of a common format but at a meta-level, the federated approach, in which no common format is defined. This approach maintains the identity of interoperating systems; nothing is imposed by one party or another and interoperability is managed in an ad-hoc manner.

The Ontology of Enterprise Interoperability (OoEI). The approach adopted for building the OoEI considers interoperability from a problem-solving perspective, not restricted to communication matters. Indeed, contrary to what can be found in most of the available definitions, interoperability is not only related to communication. The components of a system do not necessary have to communicate, but might simply have to be composed together for a specific purpose. This is illustrated by the following definition: *An interoperability problem appears when two or more incompatible systems are put in relation. Interoperability per se is the paradigm where an interoperability problem occurs* [7]. Based on an analysis on the Enterprise frameworks and existing interoperability models [12], the OoEI was defined as depicted by the figure 1.

Interoperability is implemented as a subclass of the *Problem* concept. Problems of interoperability exist when there is a relation, of any kind, between

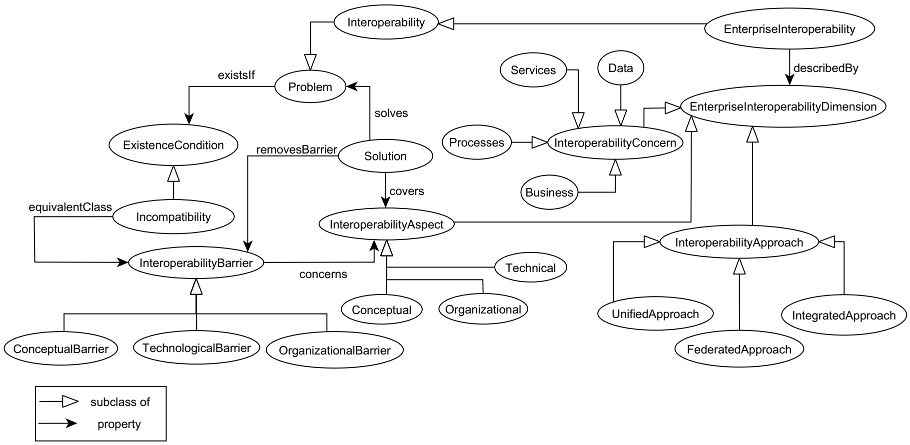


Fig. 1. Extract of OoEI

incompatible systems in a super- system they belong to or system they will form. *Incompatibility* concept is a subclass of a more generic *InteroperabilityExistence-Condition* class aiming at explicitly formalizing the fact that Incompatibility is the source of interoperability problems for systems of any nature, as soon as they belong to the same super-system and there is a relation of any kind between those systems.

Three main dimensions of EI are considered: Interoperability aspects (conceptual, organizational and technical), Interoperating entities, also known as EI concerns (i.e. business, process, service and data) and Interoperability approaches (integrated, unified and federated). These are represented by the three concepts: *InteroperabilityAs-pect*, *InteroperabilityApproach*, and *Interoperability-Concern* respectively. These are all modeled with their different constituents represented here as instances under the *EnterpriseInteroperabilityDimension* concept, as shown in figure 1.

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 [21]. It is then assimilated (with the *equivalentClass* in figure 1) to the *Incompatibility* concept. 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*.

Relevant Concepts in Enterprise Interoperability. Dealing with enterprise interoperability requires consideration of the enterprise from a general perspective, taking into account not only its different components and their interactions but also the environment in which it evolves and the interface through which it communicates with its environment.

The interface is a system's element through which a connection between the system and its environment can be established. It also represents the system's boundaries. The interfaces are important for developing interoperability and avoiding interoperability problems. An enterprise is considered as a complex system in the sense that it has both a large number of parts and the parts are related in ways that make it difficult to understand how the enterprise operates and to predict its behavior The establishment or diagnosis of enterprise interoperability leads to identify the different operational levels that are concerned. Four enterprise levels are defined in the FEI, namely business, process, service and data. They represent the areas concerned by interoperability in the enterprise.

– Interoperability of data aims to make work together different data models with different query languages to share information coming from heterogeneous systems.
– Interoperability of services aims at making it possible for various services or applications (designed and implemented independently) to work together by solving the syntactic and semantic differences.
– Interoperability of processes aims to make various processes work together. In the interworked enterprise, the aim will be to connect internal processes of two companies to create a common process.
– Interoperability of business aims to work in a harmonized way to share and develop business between companies despite the difference of methods, decision making, culture of the enterprises or, the commercial making.

According to [21], there are three kinds of interoperability problems, called also barriers that enterprises may face: conceptual, technological or organizational.

– Conceptual problems are mainly concerned with the syntactic and semantic incompatibilities of information to be exchanged or to be used during an interoperation. These problems concern the modeling at the higher level of abstraction (i.e. enterprise models) as well as the level of programming (i.e. low capacity of semantic representation of XML). Syntactic differences can be found whenever different structures are used to represent information and knowledge.
– Technological problems refer to the use of computer or ICT (Information and Communication Technologies) to communicate and exchange information (i.e. architecture and platforms, infrastructure). These problems concern the standards to use, store, exchange, processes or computerize information.
– Organizational problems relate to the definition of responsibilities and authorities so that interoperability can take place under good conditions. Responsibility needs to be defined in order to delegate tasks (process, data, software, computer). If responsibility in an enterprise is not clearly and explicitly defined, interoperation between two systems is obstructed. Authority is an organizational concept which defines who is authorized to do what. For example, it is necessary to define who is authorized to create, modify, maintain data, processes, services, etc.

3.2 Enterprise Architecture

Architecture is a consistent whole of principles, methods and models that are used in the design and realization of organizational structure, business processes, information systems, and infrastructure [22]. The use of an enterprise architecture helps to chart the complexity of an organization. The specification and description of organizations components and especially their relationships in architecture requires a coherent architecture modeling language.

Overview of EA Modeling Languages. The objective here is not to provide an exhaustive review of existing EA modeling languages but rather to present relevant modeling languages that are selected specifically for the purpose of this work. So far, the most known EA modeling languages are the Unified Enterprise Modeling Language (UEML), Design and Engineering Methodology for Organizations (DEMO) and ArchiMate.

The UEML is an on-going attempt to develop theories, technologies and tools for integrated use of enterprise and IS models expressed using different languages. By this we mean keeping the existing models as they are and establish relationships between them in an explicit and usable way, supporting, e.g., consistency checking, automatic update reflection, model-to-model translation and other services across modeling language boundaries. UEML is thus intended as an intermediate language - or a hub - through which different languages can be connected, thereby facilitating a web of languages and of models expressed in those languages [23].

DEMO is a method comprising of a comprehensive set of conceptual modeling techniques, in combination with a theory based way of thinking and associated way of working, focused on modeling, analyzing and designing the essential aspects of an organization [24]. DEMO uses the word essential here to refer to the implementation-independent aspects of an organization. As such, DEMO aims to abstract away from implementation-specific details, such as the information systems present in business collaboration [25].

ArchiMate is an Open Group standard [11] for the modeling of enterprise architectures , emphasizing a holistic view of the enterprise. This means that architects can use ArchiMate to model, amongst others, an organization's products and services, how these products and services are realized/delivered by business processes, and how in turn these processes are supported by information systems and their underlying IT infrastructure. ArchiMate is geared towards Information processing dominant organizations such as banks, insurance companies, government agencies, etc. [11].

Compared to other EA modeling languages, ArchiMate is successfully used and applied in many industrial cases [26]. Subsequently, we propose in this work to focus on ArchiMate and its main concepts. The choice of the language is motivated by the capacity of ArchiMate to distinguish between the structural or static aspect and the behavioral or dynamic aspect of enterprises. Moreover, our analysis will be limited to the ArchiMate business layer meta-model. As implied by name, the business layer focuses on an organization's business concepts such as products, (commercial) services, and business processes. The business architecture results from the implementation of business strategies and the definition of processes. The functional requirements of business process support systems, i.e. the information systems that will operationally support the business, are derived from this architecture [26].

Note that, the same approach could be applied with any other EA modeling language following the same research methodology.

ArchiMate and Its Main Concepts. In [22], three main layers are defined:

1. The Business layer offers products and services to external customers, which are realized in the organization by business processes performed by business actors.
2. The Application layer supports the business layer with application services which are realized by (software) applications.
3. The Technology layer offers infrastructural services (e.g., processing, storage and communication services) needed to run applications, realized by computer and communication hardware and system software.

Our analysis concerns the ArchiMate business layer meta-model. Figure 2 provides an excerpt of the ArchiMate business layer concepts and their relations. Note that we use only an excerpt of the ArchiMate business layer meta-model so as to focus on those concepts and relations relevant for interoperability. The ArchiMate business layer meta-model concepts, adapted from [11, 22] are described as follows:

– Business actor: It defines an individual persons (e.g., customers or employees), but also groups of people (e.g., departments or business units) within the organizations.
– Business role: A role that an actor fulfills in an organization. Importantly, this role is usually defined as the work carried out by an actor.
– Business collaboration: It defines a (temporary) configuration of two or more business roles resulting in specific collective behavior in a particular context.
– Organizational service: It is a unit of functionality that is meaningful from the point of view of the environment. The following concepts realize a service: Busi-ness processes, business functions, business interactions. Moreover, A business process/function is a unit of internal behavior, performed by one or more roles within the organization. Finally A business interaction is a unit of behavior similar to a business process or function, but it is performed in a collaboration of two or more roles within the organization.
– Business event: An event that happens (externally) and may influence business processes, functions or interactions. A business event is most commonly used to model something that triggers behavior, but other types of events are also conceivable: e.g., an event that interrupts a process.
– Business object: An entity manipulated by behavior such as business processes or functions.

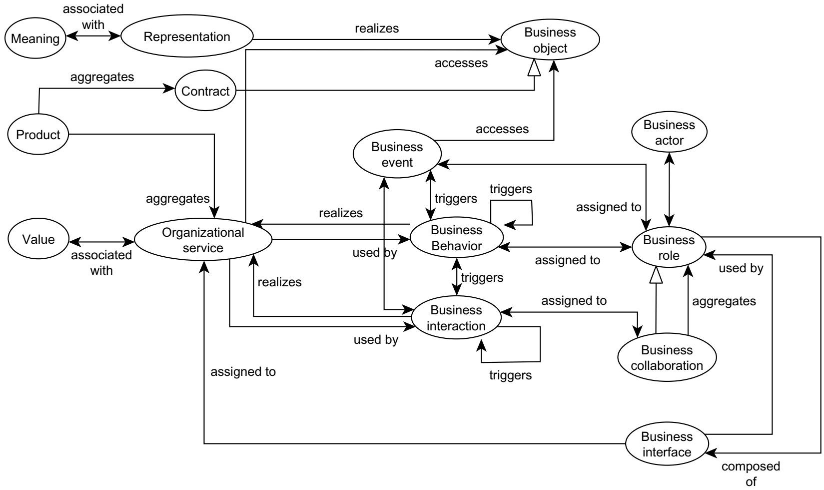


Fig. 2. ArchiMate business layer concepts

4 Conceptual Model Construction

4.1 Enterprise Architecture and Interoperability

ArchiMate, as presented in section 3.2, does not consider properly interoperability concepts. In the meanwhile, we may found concepts and relations exploring potential collaboration/interactions with local or external units to the main organization that may be a source of interoperability [5, 22]. Subsequently, this emphasizes the motivation of bridging the gap between both research areas EA and EI. Moreover, it presents tendency research topics for networked enterprises dealing with current challenges such as economic changes, globalization, et cetera. On one hand, interoperability framework needs EA models to analyze current situation and path forward future design for enterprises. On the other hand, EA models will be enriched with interoperability concepts to tackle new networked enterprises requirements (e.g., outsourcing).

Being "interoperable" refers to being able to share information between business partners, understand and process exchanged data, seamlessly integrate it into internal ICT systems, and enable its beneficial use [27]. In the following we propose the construction of the integrated model at the business layer. As implies by name, the business layer focuses on business concepts such as services, business process, product, etc. From the interoperability side, we focus also on business interoperability that can be defined as the organizational and operational ability of an enterprise to cooperate with its business partners and to efficiently establish, conduct and develop IT-supported business relationships with the objective to create values [27].

4.2 Integration of ArchiMate EA Concepts into OoEI

In this section, we define the integrated enterprise architecture interoperability model at the business layer. Figure 3 gives an overview of this model.

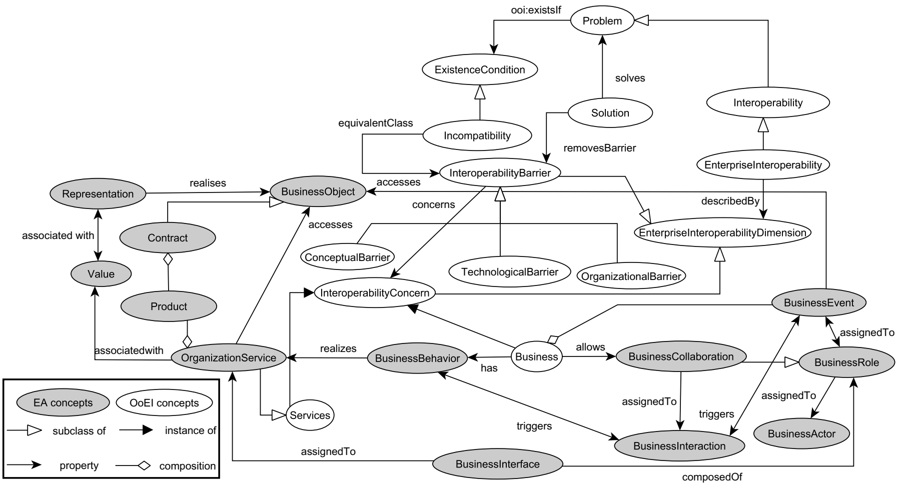


Fig. 3. Integrated Model: EA concepts and interoperability at business layer

In order to differentiate the EA concepts from the interoperability ones, we present them with the gray color.
Business collaboration, that defines a configuration of two or more business roles, will not be possible if the two roles are incompatible. Hence the importance of the interoperability that allows this kind of collaboration. This is designed by the property allows between the concepts business and *businessCollaboration*.

The two main dimensions of EI: *interoperability barriers* and *interoperability concerns* are considered. Hence we find the concepts *InteroperabilityBarrier* and *InteroperabilityConcern* as well as their respective sub-classes: *OrganizationalBarrier*, *TechnologicalBarrier*, *ConceptualBarrier* and *Business, Process, Service,* and *Data*.

The *business interoperability* concern has a behavior allowing it to realize a service. Hence the concept *Business* is related to *BusinessBehavior* by "has" property and the *BusinessBehavior* is related to *OrganizationService* by "realize" property.

5 Evaluation Using a Case Study

As part of the research approach, this section illustrates the evaluation of the proposed model using an industrial case study. The use case concerns a multinational company: METS (Manufacture Electro-Technical of Sousse) a subsidiary of the

German Draxelmaer group[1], specialized in automobile manufacturers including wiring harness systems, exclusive interiors and electrical components. To ensure its functions and reach its objectives, the company needs to interoperate with many partners, including its headquarters in Germany. This is relevant for the application of our approach.

In order to understand the way the enterprise functions, a series of interviews were conducted, our integrated model was then instantiated using the information of the company. METS is a 100% export oriented company: all its production is directed to the headquarters in Germany. The headquarters are then responsible for the distribution to the clients or other production sites. The normal business process starts when the company receives an order of production from the headquarters in Germany. If the order concerns a new product, then a prototyping is needed and a sample is produced. After a decision is reached, the production process can be launched. There are five main stakeholders for the company:

- The headquarters in Germany, from where the company receives orders.
- The production site in Poland to whom the company exports the semi-final prod-ucts.
- The production sites, from where the company receives semi-final products to finalize.
- The suppliers of the raw materials and accessories.
- Customs for the export.

As analyzing relations are the first requirement for identifying interoperations, a formal representation of the METS Company and the main relations are provided, using the integrated model. In order to differentiate between the instantiated concepts (i.e. specific to the company) and those of the proposed model, the instantiated concepts are represented by rectangle shapes as shown in figures 4 and 5.

The company is represented by the *Enterprise_Mets* concept. As an instance of *ooei:System*, it inherits all their properties and constituents. Hence it has its own structure and behavior, represented respectively by *Structure_METS* and *Behavior_METS*. The company produces wire harnesses for the cars and has two main objectives: continuous reduction of the costs of its production and to be the leader within its market. This is represented by the concept *Harness_production*, instance of *ooei:function* and two instances *ooei:objective*: *Market_leader* and *Reduce_costs*.

As any multinational enterprise, METS evolves in its environment and has many partners. This is represented by *Mets_env* concept, instance of *ooei:environment*. Within this environment, the customs, the supplier of the accessories, the transporter, the Draxelmaier headquarters, the supplier of raw material and the provider of all other services are found. This is respectively represented by the concepts: *Customs, Ac_supplier, Transporter, Drx_group,*

[1] http://www.draexlmaier.de

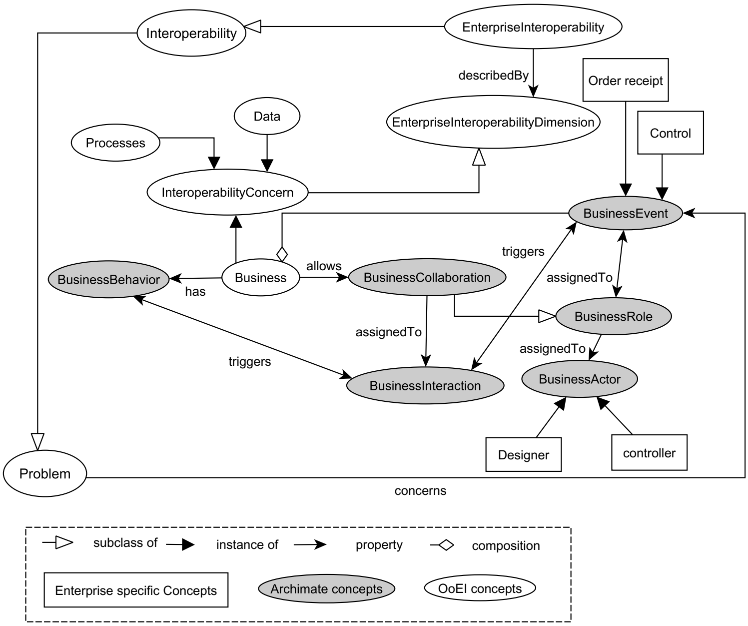


Fig. 4. Use case application - Business layer modeling

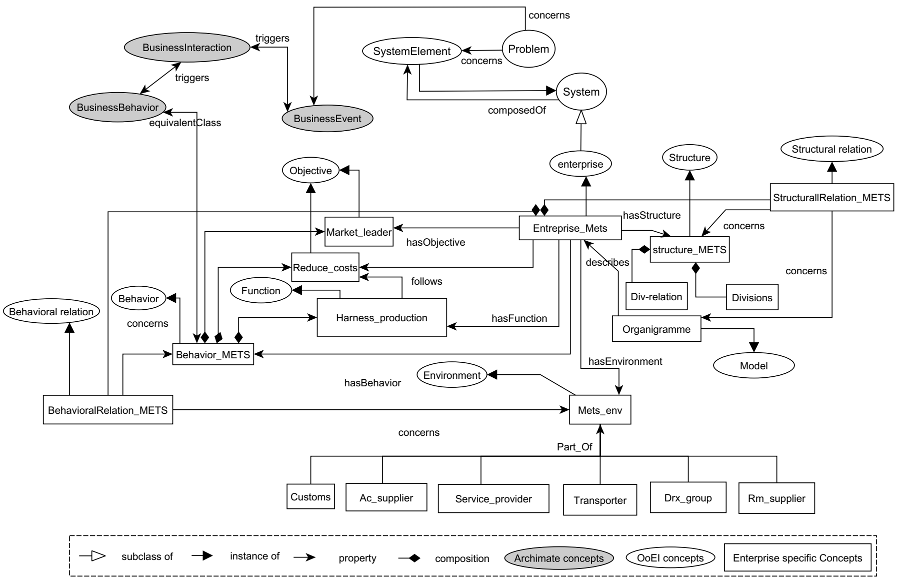


Fig. 5. Use case application- Company specification

Rm_supplier, Service_provider. The instantiation of the integrated model provides an overview of the enterprise structure and the main relations that exist.

The case study evaluation provides primer results. The maturity of this evaluation is, certainly, limited but offers an initial investigations for interoprabilty readiness in networked enterprises. Back to our scenario, we observed that based on the resulted model, the managers can have a clear idea about the main elements within their architecture when dealing with interoperability requirments. This can be thereafter extended by associated rules to allow automatic reasoning for developing tools for diagnosis of enterprise interoperability problems.

6 Conclusion and Future Work

In this paper, we have proposed an integrated model bridging the interoperability and enterprise architecture concepts following the constructive research methodology. At a first stage, the concepts were identified and defined, and then the analysis of the relations between concepts was defined in order to construct the integrated conceptual model. The evaluation stage was done through a case study. The resulted integrated model was based on the OoEI and the Archimate model at the business layer.

Future work are planned to extend the defined conceptual model with concepts from other EA modeling languages and to cover applicative and technical levels [4] of the enterprise architecture.

References

1. Dinh, T.L., Thi, T.T.P.: Information-driven framework for collaborative business service modelling. IJSSMET 3(1), 1–18 (2012)
2. Architectures for enterprise integration and interoperability: Past, present and future. Computers in Industry 59(7), 647–659 (2008)
3. Vernadat, F.: Enterprise Integration Modeling and Integration: Principles and Applications. Springer (1996)
4. Lankhorst, M.M.: Enterprise Architecture at Work - Modelling, Communication and Analysis, 4th edn. The Enterprise Engineering Series. Springer (2013) 978-3-642-29650-5
5. Institute of Electrical and Electronics Engineers. IEEE standard computer dictionary: A compilation of ieee standard computer glossaries (1990)
6. Ford, T., et al.: Measuring System Interoperability: An i-Score Improvement. In: Proceedings of the 6th Annual Conference on Systems Engineering Research (2008)
7. Naudet, Y., Latour, T., Chen, D.: A systemic approach to interoperability formalization. In: IFAC 2008 Workshops (2008)
8. Von Bertalanffy, L.: General System Theory: Foundations, Development, Applications. Georges Braziller, Inc., New York (1968)
9. Moody, D.L., Shanks, G.G.: Improving the quality of data models: empirical validation of a quality management framework. Inf. Syst. 28(6), 619–650 (2003)
10. Recker, J.: Conceptual model evaluation. towards more paradigmatic rigor. In: Proceedings CAiSE 2005 Workshops, Porto, Portugal (2005)

11. Jonkers, H., Proper, H.A., Turner, M.: TOGAF and ArchiMate: A Future Together. White Paper W192. The Open Group (November 2009)
12. Guédria, W.: A Contribution to Enterprise Interoperability Maturity Assessment. Ph.D. thesis, University of Bordeaux1, Bordeaux, France (2012)
13. Schermann, M., Böhmann, T., Krcmar, H.: Explicating design theories with conceptual models: Towards a theoretical role of reference models. In: issenschaftstheorie und gestaltungsorientierte Wirtschaftsinformatik, pp. 175–194 (2009)
14. Shanks, G., Tansley, E., Weber, R.: Using ontology to validate conceptual models. Commun. ACM 46(10), 85–89 (2003)
15. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Q. 28(1), 75–105 (2004)
16. Winter, R.: Design science research in europe. European Journal of Information Systems - Eur. J. Infor. Syst. 17(5), 470–475 (2008)
17. Vicente, P., Mira da Silva, M.: A conceptual model for integrated governance, risk and compliance. In: Mouratidis, H., Rolland, C. (eds.) CAiSE 2011. LNCS, vol. 6741, pp. 199–213. Springer, Heidelberg (2011)
18. Chen, D., Dassisti, M., Elvester, B.: Interop noe deliverable di.3: Enterprise interoperability framework and knowledge corpus - final report. Technical report, Interoperability Research for Networked Enterprises Applications and Software (INTEROP Network of Excellence), IST - Contract no.: IST-508 011 (2007)
19. AIF official website. ATHENA interoperability Framework (2004), `http://www.modelbased.net/aif/wholesite.html`
20. Miller, F.P., Vandome, A.F., McBrewster, J.: European Interoperability Framework. VDM Publishing (2011)
21. Chen, D.: Enterprise Interoperability Framework. In: Workshop on Enterprise Modelling and Ontologies for Interoperability (EMOI-INTEROP) (January 2006)
22. Jonkers, H., Lankhorst, M., Van Buuren, R., Bonsangue, M., Van Der Torre, L.: Concepts for modeling enterprise architectures. International Journal of Cooperative Information Systems 13, 257–287 (2004)
23. Vernadat, F.: Ueml: towards a unified enterprise modelling language. International Journal of Production Research 40(17), 4309–4321 (2002)
24. Dietz, J.L.G.: Enterprise ontology: theory and methodology. Springer (2006)
25. Kinderen, S., Gaaloul, K., Henderik, A.: Bridging value modelling to ArchiMate via transaction modelling, pp. 1–15 (2012)
26. Sanchez, A., Basanya, R., Ojo, A.K., Janowski, T.: Enterprise architectures - enabling interoperability between organizations. In: 8th Argentinean Symposium on Software Engineering, part of 36th Argentine Conference on Informatics, ASSE 2007, Mar del Plata, Argentina (August 2007)
27. Legner, C., Wende, K.: Towards an excellence framework for business interoperability (2006)

Pattern-based Design Research in Enterprise Architecture Management

Sabine Buckl[1], Florian Matthes[2], Alexander W. Schneider[2], and Christian M. Schweda[1]

[1] iteratec GmbH, Inselkammerstr, 4, 82008 München-Unterhaching, Germany
{sabine.buckl,christian.schweda}@iteratec.de
http://www.iteratec.de

[2] Technische Universität München,
Chair for Software Engineering for Business Information Systems,
Boltzmannstr, 3, 85748 Garching, Germany
{matthes,schneial}@in.tum.de
http://wwwmatthes.in.tum.de

Abstract. Enterprise architecture (EA) management represents an evolving discipline in the curriculum of information systems research. Research in the area of EA management is often conducted in close cooperation with industry partners, who on the one hand provide input in terms of current challenges and on the other hand are more than willing to apply and evaluate the research findings. Therefore, researchers in the area of EA management applying the design science paradigm are confronted with the challenge to make theoretical contributions which additionally can be applied and help to solve current and anticipated problems in practice.

In this paper we present the so-called pattern-based design research (PDR) an iterative design research method to overcome this problem by enabling researchers to theorize and learn from the intervention at the industry partner(s) while performing rigorous and relevant design science research. We illustrate the applicability of PDR by discussing our research projects of the last 7 years in the area of EA management. We conclude that PDR provides a suitable research method that can be applied in the area of EA management and discuss further challenges from the perspective of practitioners and researchers.

Keywords: Patterns, Design Science, Research Method, Enterprise Architecture Management, Rigor and Relevance.

1 Motivation

Enterprise Architecture (EA) Management is an evolving management discipline within the curriculum of Information Systems (IS). The discipline targets at the holistic management of the enterprise, in particular its architecture formed by elements as business processes, applications, information flows, hardware, as well as their interrelations. Researchers active in the field of EA management are

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 30–42, 2013.

faced with various challenges of high practical relevance. For these challenges, the researchers seek to develop technology-based solutions, i.e., ‘do’ design science research [1]. Such research has to be conducted in close cooperation with industry experts, in order to:

– identify a problem of practical relevance,
– develop a solution that fits the application context, and
– evaluate the purposefulness of the solution in its context.

During solution development and evaluation, design research has to rely on sound methodologies [2] to ensure rigor of the research results (cf. [3]). This puts researchers in the field of EA management research in a particularly challenging situation. On the one hand enterprises are inherently complex subjects, for which rigorous development of purposeful solutions is likely to take time. On the other hand research in close cooperation with an industry partner has to adapt to the pace of partnering organization, delivering useful results early. Therefore, researchers find themselves in a field, where – as Davenport already states in [4] – rigor and relevance compromise each other to some extent. A key challenge hence is to account for both rigor and relevance in EA management research endeavors with industry. Further, researchers are challenged to ensure that their research does not degenerate into ‘routine design’ that must be distinguished from design science [1].

EA management seeks to establish sustainable solutions to challenges that affect the structure of the enterprise as a whole. Key to this management effort is a holistic perspective on the EA. Usually, EA descriptions, containing different views on the enterprise, are used. These descriptions are again defined on a conceptual model of the enterprise. EA management uses the descriptions to increase transparency about the enterprise [5] and to control the architecture's evolution [6]. In this context, researchers target, for example, at the identification of ‘good’ architectures (behavioral science) [7] as well as at the development of models [8], methods [9], and tools [10] (design science). Especially for the latter, a close cooperation with industry is essential. Only industry experts are able to raise relevant business problems, provide data about real enterprise architectures and participate in an artifact's evaluation. To allow for rigor in a close cooperation with an industry partner which allows for relevance inherently, a design science research method has to account for several challenges occurring in such context [11]:

– Research has to provide outcomes, whose level of abstraction matches the expectations of the research community [12]. For research conducted in cooperation with industry this community is heterogeneous consisting of both practitioners and researchers, aiming at different and potentially inconsumerable levels of abstraction.
– Many EA management-related problems are “wicked problems” (cf. [13,14]). They occur, if the specific contexts of companies relate to asymmetric criteria determining different solutions [15]. To increase the relevance of research outcomes in terms of importance and suitability (cf. [16]), the company's

context has to be taken into account when performing EA management research with an industry partner.

In this paper, we describe the research method of *Pattern-based Design Research* (PDR). This method uses patterns to provide early and relevant results to practitioners and design theories to rigorously abstract these results to scientific contributions. We repeatedly applied this method to different problem dimensions in EA management, like the management processes and the description techniques. Latter application is described in more detail in this paper to show by example, how PDR can facilitate EA management research. Section 2 introduces the basic conception of PDR, referencing relevant groundwork from both pattern research and design research. The course of a research as well as the results thereof are described in Section 3 starting with early results on patterns and concluding with their abstraction to building blocks. In Section 4 we discuss challenges from both academic and practitioner direction, that we had to face applying the PDR method. Further, we give an outlook on possible future applications of PDR to EA management and briefly sketch the research ecosystem that would be needed to facilitate PDR in the context of EA management.

2 The Pattern-based Design Research Method

The Pattern-based Design Research (PDR) method introduced in [11] and visualized in Figure 1 provides a research approach for a close cooperation of researches and industry experts suitable for the context of EA management.

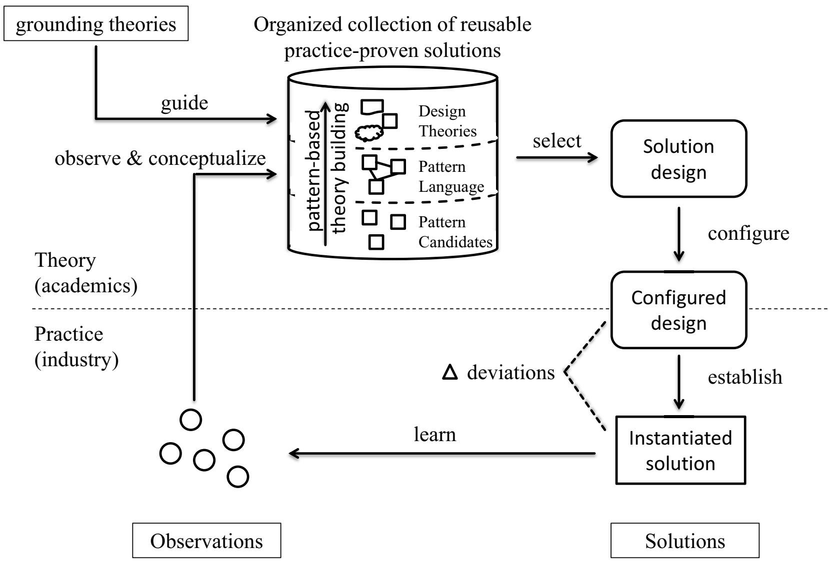


Fig. 1. The PDR method [11]

By the observation of common solutions to reoccurring EA management problems in practice and their conceptualization the basis for an EA management Design Theory Nexus (DTN) is formed. The identified patterns (cf. [17,18,19]) as well as related grounding theories are then used to build design theories [20,21]. By the conceptionalization of the context in which a pattern is suitable as well as the specific goals, a DTN is built. Industry partners willing to use the practice-proven knowledge provided by the DTN can design their organization-specific EA management function by using their context and goals as input. The resulting solution design can then be applied to the organization's context. For example, the *solution design* can be configured by adjusting the used terminology. The resulting *configured solution* serves as a blueprint to be implemented by the organization. The implementation includes the realization of processes as well as the provision of tool support if required. After the actual implementation of the configured solution we speak of an *instantiated solution*. This differentiation allows, for example, to distinguish a process's design from its instances. Of course, the instantiated solution is subject to evolution when the organization runs it over time. If process instances differ from their original design, a better solution has been implemented by the organization itself. This solution might, for example, suite better to the organizations context than the solution design created by the use of the DTN. In general, the derivations between the configured and the instantiated solution can be used as a basis to extend the knowledge contained in the DTN. In addition to adjustments of already known solutions, even completely new solutions can be observed this way. Such an observation closes the cycle of the PDR method and a new iteration begins by the solution's conceptualization to form a pattern candidate.

3 Applied PDR – Course of Action

Our research on Software Cartography, EA management patterns, and Building Blocks for EA management Solutions has followed the PDR method since 2004. Close cooperation with more than 30 industry partners over the years has allowed us to identify, design, refine, and validate many relevant solutions. Being at least partially industry-funded, the patterns as well as tooling based on the patterns has helped us to deliver useful solutions to our partners. In the following, we present the different stages of our research and outline both directly applicable results and the theoretic underpinnings developed by us. Table 1 gives an overview of the different research artifacts.

In 2004 we started a research project on visualizing application landscapes. This project was called "software cartography" [22] and could, due to the support of Prof. Dr. Denert, start with a series of interviews with practitioners. In these interviews, we identified the challenge of missing and not structured techniques for visualizing application landscapes. We further were able to collect exemplary visualizations that some of the partnering enterprises created without applying re-usable techniques. In the publications [23,24], we identified four types of "software maps", called "cluster map", "process support map",

Table 1. PDR for EA management – course of action

Year	Applicable results	Theoretic underpinnings
2004	Software Map types (1)	
2005	Software Map types (2)	IEEE 1471 for Software Cartography
2006	Software Cartography tool	Transformation perspective on Software Cartography (1)
2007		Transformation perspective on Software Cartography (2)
2008	EA management Pattern Catalog Wiki	Structure and style of EA management patterns
2009	Workshop: Patterns in EA management	
2010	Building Blocks for EA management Solutions (BEAMS)	Refining the ISO 42010 Meta-Language for EA Information Modeling
2011	BEAMS Wiki	Development of Organization-Specific EA Modeling Languages using Building Blocks

"time-interval map", and "graph-layout map". For all these types we not only described typical application scenarios, but also outlined best-practices applying to their practical application. Without having them called so explicitly, we documented the first *pattern candidates*. These pattern candidates provided the basis for continuing research in cooperation with almost 30 industry partners over the next five years.

From a more theoretic perspective, we experienced a significant lack of means in describing and formalizing how the different visualizations in an EA description interrelate. From work with the practitioners we learned that different roles in an organization are using the different visualizations for different purposes, i.e., to answer different relevant questions. The theoretical underpinnings from the field of software engineering, more precisely the "documentation of software-intensive systems" (cf. IEEE Std. 1471 [25]) provided a terminology to structure and formalize the pattern candidates. In [26] we explored software cartography as a means to document software-intensive enterprises and motivated the concepts *concern* and *stakeholder* related to the *viewpoints*, as which we understood the software map types.

Giving account to the importance of EA modeling and data collection we focused our research in the years of 2005 and 2006 on issues related to the automated creation of EA visualizations. Thereby, we identified issues in visualization handling during an extensive survey of existing tools for EA management [27]. To overcome the error-prone and time-consuming task of manual creation of EA viewpoints, we propose a model transformation approach, which links the data to be visualized and their graphical representations. The fundamental principles of software cartography, an approach for EA modeling, including a method for the automatic creation of visualizations based on EA models is presented in [28,29].

In [30] we sketch an approach to designing organization-specific information models for EA management based on patterns. This approach is intended to manage the complexity of such models, to which the wide spread domains (e.g. processes, technical architecture, strategic issues) that are involved in EA management contribute. Our contribution in this respect lies in the introduction of patterns into the field of EA management information models. Thereby, other approaches, as e.g. structuring the information models into layers, are complemented by the possibility to reuse pre-existing solutions to address EA management issues. Having provided an initial example, how such a pattern could look like, we embarked to describing more patterns, as we have encountered them in our industry projects.

Extending the idea of patterns to the area of EA management from a comprehensive perspective, we introduce three different types of EA management patterns: Information Model Patterns (I-Pattern), Viewpoint Patterns (V-Patterns), and Method Patterns (M-Patterns) [31]. M-Patterns document proven-practice methods to address typical problems in EA management. V-Patterns represent best-practice EA visualizations, and I-Patterns indicate information requirements for EA management. The EA management patterns thereby follow the typical structure of patterns (cf. [32]) containing e.g. a name, context, problem, solution, and consequence descriptions. The different EA management patterns are interlinked as e.g. a V-Pattern is referenced within the method description of an M-Pattern and the V-Pattern links to the I-Pattern defining the information demands underlying a respective V-Pattern. Thus, the different patterns build up a pattern language for EA management.

The EA management pattern catalog [33] documents EA management patterns for management methods, viewpoints, and information models as well as the dependencies between these patterns and the thereby addressed concerns. These patterns are presented in a consistent terminology and information organization to simplify the selection, adaptation, and integration of patterns. The catalog incorporates the findings from earlier research in the project Software Cartography and the EA management Tool Survey 2005 [27]. Subsequently the patterns were evaluated using an on-line questionnaire to identify methodologies and viewpoints that are considered relevant and useful by practitioners. The EA management Pattern Catalog tries to find a balance between a green field approach and a completely predefined approach as provided by some EA management frameworks and EA management tools. It avoids a giant integrated information model but utilizes a consistent terminology and a common organization to permit an understanding and comparison of multiple approaches from different sources. Sample viewpoints help readers to grasp essential concepts.

With the first workshop on “Patterns in Enterprise Architecture Management” (PEAM 2009) we seek to set the stage for EA management patterns also in the scientific community. The introductory words in [34] refrain the growing interest in academia and industry to “identify, collect, document, and exchange best-practices” for EA management. The PEAM workshop addresses European EA management researchers and practitioners to exchange their knowledge using patterns as means of documentation. An EA management pattern “is a general, re-usable solution to a common problem in a given context which identified driving forces, known usages, and consequences”. Six paper presented in the workshop address various topics ranging from processes for EA management, over modeling of smart enterprise networks, to viewpoint patterns for EA management in the context of mergers and acquisitions. The specialized format of the workshop ensured relevance and applicability of the presented solutions. During the review both scientists and practitioners were invited to judge relevance and contribution of the solution. Afterwards, a “pattern shepherd” was assigned to the authors of each accepted paper, providing guidance on the writing style of the paper. In the workshop, no comprehensive presentation was given by the authors, but well-prepared attendants discussed aspects of understandability and presentation. Their feedback was incorporated into the final papers to ensure that other readers of the patterns are able to understand and apply the presented solutions.

Focusing again more on the theoretic underpinnings of EA and EA management, we further investigated architectural descriptions in 2010. Architectural descriptions representing and modeling the architecture of a system or parts thereof are typically used in the engineering disciplines to plan, develop, maintain, and manage complex systems. Primarily originating from construction engineering, the means of architecting and architectural descriptions have been successfully transferred to related disciplines like software engineering. While a rich and formal theory on conceptual modeling exists as well as frameworks on how to approach architectural descriptions, e.g. the ISO standard 42010 [35], only few attempts have yet been made to integrate the prescriptions and guidelines from these sources into a formal architectural description framework. In [36], we establish such a framework against the background provided by the ISO standard 42010 [35] by formally defining the terms concern, view, viewpoint, and architectural description.

In the light that different EA descriptions are based on viewpoints that again entail different underlying information models, we analyze these information models in [37]. In particular, we focus on the meta-languages used for prevalent information models and diagnose their diversity. For the analysis we apply the work of Guizzardi [38], who established a theoretic base for ontological modeling. Based on the analysis of the used meta-languages, we elicit requirements for a “unifying meta-language”. The requirements for these meta-language in turn are based on recurring phenomena in EA information modeling, e.g. “hierarchy modeling” for element hierarchies or “multi-level modeling” for ontological instantiation of elements. We further reflect that multi-purpose modeling facilities, like the UML, fail to satisfy these requirements. The requirements further shape the path for subsequent theoretic research on how to describe EA information models.

The enduring need for applicable standards in the field of EA management and the lack of a commonly accepted understanding of scope, reach, and focus of EA management motivate our work in [39]. We reflect on the different organizational structures that raise implications on EA management and demand an enterprise-specific realization of the management function. We further reflect that pattern, while providing re-usable solutions, lack guidance on their integration. In response to these deficiencies we describe the building blocks for EA management solutions (BEAMS). BEAMS provides practical guidance for organizations to support the design and development of an organization-specific EA management function by presenting method and language building blocks. These building blocks are selected based on the concerns, goals, and context of the using organization. They are then integrated following structured integration procedures based, e.g. on the technique of method engineering [40,41]. Alongside we discuss how BEAMS gives hints for researchers willing to contribute to the discipline of EA management, and outline the theoretic foundations of BEAMS.

To ensure that the best practices contained in BEAMS are available to a wide audience of practitioners, the organized library of building blocks was made available using an enterprise 2.0 system[1]. In the BEAMS wiki each building block is documented by a single wiki page. Furthermore, the pages are equipped with hybrid templates that describe the characteristics of the corresponding building block in a structured manner [42].

The thesis [43] presents the complete method for developing organization-specific EA modeling languages to support EA management addressing the challenges of enterprise transformation. The method builds on a library of language building blocks resulting from the PDR over the years. These building blocks are organized according to a conceptual framework of the EA. We distinguish three types of building blocks: “information model building blocks” (IBB) describing the abstract syntax, “viewpoint building blocks” (VBB) describing the applicable concrete syntax, and “glossary building blocks” (GBB) providing textual semantic definitions for the concepts. The method further builds on a set of formalisms to integrated IBBs into a comprehensive and organization-specific information model. Additional formalisms help to apply the VBBs on the information model and to evolve the GBBs into a terminology aligning to the terms used in the organization. All formalisms are based on the theoretic underpinnings of two domain-specific meta-languages, one for information modeling and one for viewpoint definition. In addition, we provide guidelines how to consistently evolve and adapt existing EA description languages developed by this method.

4 PDR – Experiences, Challenges and Outlook

In this paper, we outlined our Pattern-based Design Research (PDR) method and their successful application to the field of EA management, in particular EA modeling. We experienced both the benefits of applying the method as well as challenges in application, which we describe subsequently.

[1] The BEAMS wiki is available at
`http://wwwmatthes.in.tum.de/pages/ste22z023rd3/BEAMS`
(last accessed 2013-04-03).

The practice-driven research and the early delivery of results as patterns was highly appreciated by the industry partners supporting and funding our research. In particular, the viewpoint patterns described in the pattern catalog have been applied as means to determine, which 'images' of the architecture would be useful to have. A number of (invited) talks on the topic [44,43] and several publications in 'yellow press' [45,46] reflect the positive reaction of practitioners regarding the EA management patterns. The pattern catalog wiki further promoted the findings and more than 1000 registered users accessing the patterns in 2010 (2 years after setup of the wiki) give an indication on the practical impact of the findings [47].

Nevertheless, these registered users did not form the active community that we expected them to form. We enabled the users to comment on patterns and to document known uses in practice. Further, we invited users with a track-record in EA management research and practice to contribute their findings and results as patterns. After two years, only a small number of comments (less than 20) were added to the wiki, and in addition to the patterns received at the PEAM workshop only one pattern was documented by an invited user. This in particular ascribes to the fact that EA management practitioners in a single company have difficulties in observing the same pattern repeatedly, i.e., three times. Practitioners with the possibility to observe solutions at different organizations, e.g. consultancies, on the other hand are not that likely to share their overarching insights in the field.

In publishing the patterns that we observed, identified, and documented in our research we experienced several difficulties. In early publications in the IS environment, reviewers were susceptible to the idea of applying patterns in the field of EA management. A similar discussion on the applicability of pattern in the context of IS research can be found in [48]. When it came to the presentation of single patterns, outlets with a strong focus on rigorous research were not likely to accept, what the reviewers called "non-representative cases". Different reviewers advised us to perform empiric analyses, but thereby did not acknowledge the fact that empiric analyses targeting novel design solutions are hard to undertake. Pattern conferences and workshops, like the PLoP conference series[2] conversely provided an outlet receptive for pattern-based documentation of results. The feedback received in the reviews and on-site nevertheless did not focus on the relevance and applicability of the solution, but on its documentation, i.e., the 'patternness' of its documentation. In our opinion nevertheless, feedback on all three facets of a pattern solution would be required, i.e., feedback on:

- the practical *relevance* of the presented context, problem, and solution,
- the *theoretic groundwork*[3] on which context, problem, and solution description rely, and
- the *patternness* of the documentation, i.e., the understandability to a practitioner.

[2] See `http://hillside.net/conferences` (last accessed 2014-04-03).
[3] Grounding theories.

Our PEAM workshops were an attempt to ensure that feedback regarding all three facets can be provided. A program committee of both practitioners and researchers in the particular field review relevance of the pattern and appropriateness of the theoretic foundations. Once a pattern is regarded relevant and sufficiently grounded, a shepherd supports the authors to improve the pattern-like presentation. This presentation in turn is subjected to a final discussion at the writers' workshop, at which practitioners and researchers provide final feedback. With the novel format of the workshops, we were able to overcome what we experienced to be deficiencies of 'classic' publication outlets, when it comes to patterns.

The PDR method has proven its applicability in the context of EA management, but is in its nature more universally applicable. Disciplines and research fields with high practical relevance, wicked design problems, and the need to perform research in cooperation (and with the funding) of practitioners are likely susceptible for this research approach. Possible fields may be design solutions for cloud computing, enterprise-wide knowledge management, or business intelligence. The method does further not rely on a completely open presentation of the pattern candidates and patterns: while a receptive and critic external audience promotes the formulation of relevant solutions, also consultancies may take advantage from applying the PDR method. Pattern candidates and patterns can in this context be useful to document the findings from different consulting projects, and to consolidate the relevant knowledge in a knowledge base for the organization.

References

1. Hevner, A., March, S., Park, J., Ram, S.: Design science in information systems research. Management Information Systems Quarterly 28(1), 75–105 (2004)
2. Benbasat, I., Weber, R.: Rethinking 'diversity' in information systems research. Information Systems Research 7, 389–399 (1996)
3. Gallupe, R.B.: The tyranny of methodologies in information systems research. SIGMIS Database 38, 20–28 (2007)
4. Davenport, T.H., Markus, L.M.: Rigor and relevance revisited: Response to benbasat and zmud. Management Information Systems Quarterly 23(1), 19–23 (1999)
5. Hanschke, I.: Strategic IT Management: A Toolkit for Enterprise Architecture Management. Springer (2009)
6. Murer, S., Bonati, B., Furrer, F.J.: Managed Evolution. Springer (2010)
7. Niemi, E., Pekkola, S.: Enterprise architecture quality attributes: A case study. In: 46th Hawaii International Conference on System Sciences (HICSS), pp. 3878–3887 (2013)
8. Frank, U.: Domain-specific modeling languages – requirements analysis and design guidelines. In: Reinhartz-Berger, I., Sturm, A., Clark, T., Wand, Y., Cohen, S., Bettin, J. (eds.) Domain Engineering: Product Lines, Conceptual Models, and Languages. Springer (2013)
9. Riege, C., Aier, S.: A contingency approach to enterprise architecture method engineering. In: Pre-Proceedings of the 3rd Workshop on Trends in Enterprise Architecture Research (TEAR 2008), Sydney, Australia (2008)

10. Matthes, F., Neubert, C., Schneider, A.W.: Fostering collaborative and integrated enterprise architecture modelling. Enterprise Modelling and Information Systems Architectures 8(1) (2013)
11. Buckl, S., Matthes, F., Schneider, A.W., Schweda, C.M.: Pattern-based design research – an iterative research method balancing rigor and relevance. In: Proceedings of the 8th International Conference on Design Science Research in Information Systems and Technology (2013)
12. Vaishnavi, V., Kuechler, W.: Design research in information systems (2004), `http://desrist.org/design-research-in-information-systems`
13. Churchman, C.W.: Guest editorial: Wicked problems. Management Science 14(4), B141–B142 (1967)
14. Rittel, H., Webber, M.: Dilemmas in a general theory of planning. Policy Sciences 4, 155–169 (1973)
15. Pries-Heje, J., Baskerville, R.: The design theory nexus. Management Information Systems Quarterly 32(4), 731–755 (2008)
16. Rosemann, M., Vessey, I.: Toward improving the relevance of information systems research to practice: the role of applicability checks. Management Information Systems Quarterly 32(1), 1–22 (2008)
17. Alexander, C.: Notes on the Synthesis of Form. Harvard University Press, Cambridge (1972)
18. Gamma, E., Helm, R., Johnson, R., Vlissides., J.M.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional (1994)
19. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M.: Pattern-oriented software architecture: a system of patterns. John Wiley & Sons, Inc., New York (1996)
20. Buckl, S., Matthes, F., Schweda, C.M.: Utilizing patterns in developing design theories. In: 2010 International Conference on Information Systems (ICIS 2010) (2010)
21. Buckl, S., Matthes, F., Schweda, C.M.: From ea management patterns towards a prescriptive theory for designing enterprise-specific ea management functions – outline of a research stream. In: Multikonferenz Wirtschaftsinformatik (MKWI 2010), Gttingen, Germany, pp. 21–22 (2010)
22. sebis: Jahresbericht lehrstuhl für informatik 19, `http://wwwmatthes.in.tum.de/file/1md32csjj4emf/sebis%20Public %20Website/Publications/sebis--Ganzjahresbericht-- 2002--2004.pdf (2004)` (last accessed April 3, 2013
23. Matthes, F., Wittenburg, A.: oftwarekartographie: Visualisierung von anwendungslandschaften und ihrer schnittstellen. In: Informatik 2004 - Informatik verbindet, 34. Jahrestagung der GI, Ulm (2004)
24. Lankes, J., Matthes, F., Wittenburg, A.: Softwarekartographie: Systematische darstellung von anwendungslandschaften. In: Ferstl, O., Sinz, E., Eckert, S., Isselhorst, T. (eds.) Wirtschaftsinformatik 2005, pp. 1443–1462. Physica-Verlag (2005)
25. IEEE: Recommended Practice for Architectural Description for Software-Intensive Systems. IEEE Std. 1471-2000 (2000)
26. Lankes, J., Matthes, F., Wittenburg, A.: Softwarekartographie als beitrag zum architekturmanagement. In: Aier, S., Schnherr, M. (eds.) Unternehmensarchitekturen und Systemintegration, vol. 3. Gita, Berlin (2005)
27. Matthes, F., Buckl, S., Leitel, J., Schweda, C.: Enterprise Architecture Management Tool Survey 2005. TU Mnchen, Chair for Informatics 19, Prof. Matthes (sebis) (2005)

28. Ernst, A.M., Lankes, J., Schweda, C.M., Wittenburg, A.: Using model transformation for generating visualizations from repository contents – an application to software cartography. Technical report, Chair for Informatics 19 (sebis), Technische Universitt Mnchen, Munich, Germany (2006)
29. Buckl, S., Ernst, A.M., Lankes, J., Matthes, F., Schweda, C.M., Wittenburg, A.: Generating visualizations of enterprise architecture using model transformation (extended version). Enterprise Modelling and Information Systems Architectures – An International Journal 2(2), 3–13 (2007)
30. Buckl, S., Ernst, A., Lankes, J., Schneider, K., Schweda, C., Wittenburg, A.: A pattern based approach for constructing enterprise architecture management information models. In: 8. Internationale Tagung Wirtschaftsinformatik, Karlsruhe, Germany, pp. 145–162 (2007)
31. Ernst, A.: Enterprise architecture management patterns. In: Pattern Languages of Programs Conference 2008 (PLoP 2008), Nashville, USA (2008)
32. Meszaros, G., Doble, J.: A Pattern Language for Pattern Writing, pp. 529–574. Addison Wesley (1998)
33. Buckl, S., Ernst, A.M., Matthes, F.: Enterprise architecture management pattern catalog. Technical report, Technische Universitt Mnchen, Munich, Germany (2008)
34. Matthes, F., Ernst, A.: Patterns in enterprise architecture management (peam). In: Münch, J., Liggesmeyer, P. (eds.) Software Engineering 2009 – Workshopband, Bonn, Germany. Lecture Notes in Informatics (LNI), pp. 17–18 (2009)
35. International Organization for Standardization: ISO/IEC 42010:2007 Systems and software engineering – Recommended practice for architectural description of software-intensive systems (2007)
36. Buckl, S., Krell, S., Schweda, C.M.: A formal approach to architectural descriptions – refining the iso standard 42010. In: Albani, A., Dietz, J.L.G. (eds.) CIAO! 2010. LNBIP, vol. 49, pp. 77–91. Springer, Heidelberg (2010)
37. Buckl, S., Matthes, F., Schweda, C.M.: A meta-language for ea information modeling - state-of-the-art and requirements elicitation. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Ukor, R. (eds.) BPMDS 2010 and EMMSAD 2010. LNBIP, vol. 50, pp. 169–181. Springer, Heidelberg (2010)
38. Guizzardi, G.: Ontological Foundations for Structural Conceptual Models. Telematica Instituut, Enschede, The Netherlands (2005)
39. Buckl, S., Dierl, T., Matthes, F., Schweda, C.: Building blocks for enterprise architecture management solutions. In: Harmsen, F., Proper, E., Schalkwijk, F., Barjis, J., Overbeek, S. (eds.) PRET 2010. LNBIP, vol. 69, pp. 17–46. Springer, Heidelberg (2010)
40. Brinkkemper, S.: Method engineering: engineering of information systems development methods and tools. Information and Software Technology 38(4), 275–280 (1996)
41. Harmsen, A.F.: Situational Method Engineering. PhD thesis, University of Twente, Twente, The Netherlands (1997)
42. Matthes, F., Neubert, C.: Wiki4eam: Using hybrid wikis for enterprise architecture management. In: 7th International Symposium on Wikis and Open Collaboration (WikiSym), Mountain View, California (2011)
43. Schweda, C.M.: Development of Organization-Specific Enterprise Architecture Modeling Languages Using Building Blocks. PhD thesis, Technische Universitt Mnchen (2011)
44. Buckl, S.M.: Developing Organization-Specific Enterprise Architecture Management Functions Using a Method Base. PhD thesis, Technische Universität München, München, Germany (2011)

45. Buckl, S., Ernst, A.M., Schweda, C.M.: Eam patterns: Best practices for enterprise architecture management. Architecture & Governance Magazine 5 (2009)
46. Buckl, S., Ernst, A.M., Lankes, J., Matthes, F.: Eam pattern catalog liefert best practices. isReport 5 (2008)
47. Ernst, A.M.: A Pattern-Based Approach to Enterprise Architecture Management. PhD thesis, Technische Universität München, München, Germany (2010)
48. Winter, R., Vom Brocke, J., Fettke, P., Loos, P., Junginger, S., Moser, C., Keller, W., Matthes, F., Ernst, A.M.: Patterns in business and information systems engineering. Business & Information Systems Engineering, 468–474 (2009)

Developing the GEA Method – Design Science and Case-Study Research in Action*

Roel Wagter[1,4], Hend*erik* A. Proper[2,4], and Dirk Witte[3,4]

[1] PMTD, Rotterdam, The Netherlands
[2] CRP Henri Tudor, Luxembourg
[3] Logica, Amstelveen, The Netherlands
[4] Radboud University Nijmegen, Nijmegen, The Netherlands
roel.wagter@pmtd.nl, erik.proper@tudor.lu, dirk.witte@logica.com

Abstract. This paper is concerned with the research methodology that was used in the GEA (General Enterprise Architecting) research programme. The goal of this programme was the development of a new approach for doing enterprise architecture. We discuss the motivations for starting the GEA programme, its focus, as well as its objectives. Based on this, the research methodology as it was used by the GEA programme is discussed and motivated. This involves a combination of design science and case study based research. The paper also discusses the way the GEA programme went about to actually implement the research methodology in a real-world situation, while also highlighting its results.

Keywords: Enterprise architecture, enterprise coherence, design science.

1 Introduction

This paper focuses on the research methodology used in the GEA (General Enterprise Architecting) research programme. The goal of this programme was the development of a new approach to enterprise architecture. The scientific results of this programme have been reported in [9, 17, 18, 19, 20, 21, 22]. The focus of this paper is not on a discussion of these results, but rather on the way the programme was structured, the research methodology used, and the way it was actually implemented in practise.

In Section 2 we will briefly discuss the motivations for the GEA programme, its focus, and its objectives. This provides us with the context to discuss, in Section 3, the research methodology used. Section 4, then reports on the way this was actually implemented during the execution of the GEA programme. Finally, before concluding, Section 5 highlights the results of the research programme.

2 The GEA Research Programme

The GEA method was developed in a multi-client research programme involving twenty organizations: ABN-AMRO Bank; ANWB (Dutch Automobile Club); Achmea

* This work has been partially sponsored by the *Fonds National de la Recherche Luxembourg* (www.fnr.lu), via the PEARL and CORE programmes.

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 43–57, 2013.

Insurances; Belastingdienst – Centrum voor ICT (IT department of the Dutch Tax Authority); ICTU (IT expertise centre of the Dutch government); ING Bank; Kappa Holding; Ministerie van Binnenlandse Zaken en Koninkrijksrelaties (Ministry of Foreign Affairs); Ministerie van Defensie (Ministry of Defence); Ministerie van Veiligheid & Justitie – Dienst Justitiële Inrichtingen (Ministry of Public Safety and Justice); Ministerie van Landbouw, Natuur en Voedselkwaliteit (Ministry of Agriculture, Nature and Food quality); Nederlandse Spoorwegen (Dutch Railways); PGGM Pension Fund; Politie Nederland (Dutch Police); Prorail; Provincie Flevoland; RaboBank; Rijkswaterstaat (Executive agency of the Dutch Ministry of Infrastructure and the Environment); UWV (Agency for work & social well-fare); Wehkamp.

The development of GEA was initiated in 2007 by the consultancy firm Ordina. The core driver for them to initiate the GEA programme [13] was their observation that in their experience enterprise transformations (changing an enterprise from its business processes to its underlying IT) fail more often than not. They also observed how existing methods and frameworks for enterprise architecture fell short in contributing to the success of enterprise transformation efforts [9, 13].

These experiences were also supported by a study conducted by the (Dutch) *General Court of Auditors* on the causes of failures of IT projects in the public sector in the Netherlands [1]. The resulting report [1] also links the failure of these projects to a lack of coherence/alignment between core aspects of the involved government agencies (translated from Dutch):

"*ICT projects for the government seem to be much more expensive than anticipated initially, require more time than planned to complete, or do not deliver the desired results. This is a serious matter, since ICT projects of the government mostly involve the spending of public money. Furthermore the effects of projects that fail, to a larger or lesser extent, are often large-scale projects with profound social impact.*

The most important cause of the (partial) failure of ICT projects revealed by the first part of the research was that ICT projects for the government are often overly ambitious and too complex because of the combination of politics, organizational and technical factors. With these overly complex projects there is no balance between ambition, available people, resources and time."

A survey [13] held at the start of the GEA research programme showed that these experiences were also shared among the client organizations participating in the programme. The underlying issues were considered grave enough for these organizations to indeed co-invest, in terms of time and money, in the GEA research programme. For the involved organizations, the important triggers were:

- many enterprise transformation efforts fail,
- failing to adopt a holistic approach to address key business issues, frequently leading to a unilateral approach from an IT oriented perspective,
- existing architecture methods fall short in meeting their promises because:
  - they are set up from an IT perspective only,
  - they hardly address the strategic level of the organization,
  - they are set up in terms of the Business/IT gap and
  - their underlying IT architectures applied on the enterprise-wide level are unjustly called enterprise architectures.

The aforementioned survey [13] was also used to the requirements on the outcomes of the research programme. This also resulted in the formulation of the driving postulate of the programme: *the overall performance of an enterprise is positively influenced by a proper coherence among the key aspects of the enterprise, including business processes, organizational culture, product portfolio, human resources, information systems, IT support, etc.* The GEA programme refers to the latter coherence as enterprise coherence [13]. Enterprise coherence subsumes the traditional notion of 'Business-IT alignment' in the sense that it is not just 'business' and 'IT' that need to be aligned [17].

The above mentioned triggers were discussed in a workshop with a customer reference group of the client organizations involved in the GEA programme [13], leading to the formulation of the following problem definition: *Many organizations lack enterprise coherence governance*, where *enterprise coherence* was defined as follows [18]: "*Enterprise coherence is the extent to which all relevant aspects of an enterprise are connected, in such a way that these connections facilitate an enterprise obtaining/meeting its desired results*". Based on this five key research questions were formulated:

1. *What are the core factors that define enterprise coherence?*
2. *What are the core factors that influence enterprise coherence?*
3. *What impacts does the governance of enterprise coherence have on the performance of enterprises in practise?*
4. *How can enterprise coherence be expressed explicitly?*
5. *How can enterprise coherence be governed?*

More specifically, the research objectives of the research programme were:

1. *Definition of the core indicators and factors that define enterprise coherence.*
2. *Definition of the core indicators and factors that influence enterprise coherence.*
3. *Identification of the potential impact factors of enterprise coherence governance on the organizational performance.*
4. *Be able to measure an enterprise's level of coherence governance.*
5. *A design theory on how to guard/improve the level of coherence in enterprises.*

The latter design theory is referred to as the *GEA method.*

3 Research Methodology

3.1 Design Science as the Basic Approach

In developing the GEA method, the GEA programme used the Design Science research methodology. Design science addresses research through the building and evaluation of artefacts designed to meet the identified business need [4]. Design science research can be characterized as prescriptive research because it focuses on using existing knowledge to improve the performance of systems [6]. Thus, artefacts resulting from Design Science research are geared towards addressing business or organizational needs in a problem domain [4], or offering opportunities of improving practise even before practitioners identify any problem with their way of working [5]. Figure 1 shows how the Design Science approach has been applied in the GEA programme.

The box on the left-hand side of Figure 1 represents the problem domain of the research programme, i.e. the environment of enterprise coherence governance consisting of organizations in the public and industrial area with more than 200 employees, a high degree of multiple forms of labour division, the business issues that influence the level of coherence and the people involved in enterprise coherence governance.

The boxes in the middle of Figure 1 represent the two major phases of the research programme, i.e. the develop/build phase and the evaluation phase of the intended theory and artefacts (i.e. GEA). The box on the right-hand side of Figure 1 shows examples of the theories, frameworks, instruments, constructs, models, techniques, measures and validation criteria that were adopted to develop GEA, so that it supports the execution of enterprise coherence governance. The Figure also shows that the GEA theory/method actually consists of three components (artefacts in design science terminology): ECA (Enterprise Coherence-governance Assessment) [17, 22], ECF (Enterprise Coherence Framework) [18] and ECG (Enterprise Coherence Governance-approach) [19, 23].

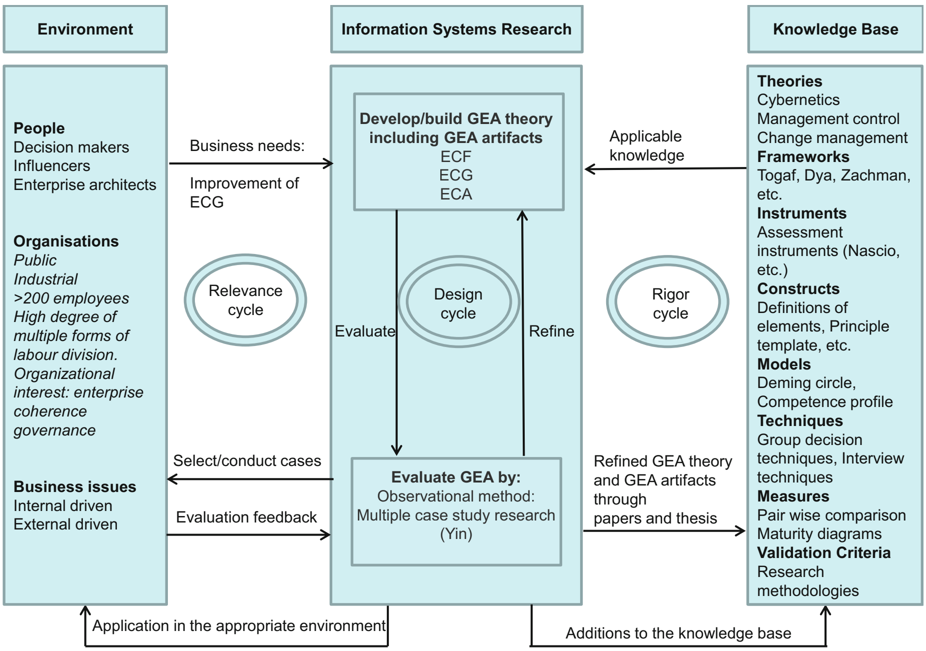


Fig. 1. Adoption of Design Science approach by the GEA programme [3, 4]

3.2 Adoption of Design Science Research Guidelines

Hevner et al [4] provide seven guidelines for proper Design Science research. Below we highlight how the GEA programme has endeavoured to meet these guidelines.

Guideline 1: Design as an Artefact. Design science research must produce a viable artefact in the form of a construct, a model, a method, or an instantiation [4].

The key artefact resulting from the research programme is the GEA method, consisting of a theory and the artefacts ECF, ECG and ECA.

Guideline 2: Problem Relevance. The objective of Design Science research is to develop '*technology-based solutions*' to important and relevant business problems [4]. The problem environment mainly comprises the level of purpose and the level of design of organizations, the business issues that influence the coherence at and between these levels and the people involved in enterprise coherence governance. The people involved in enterprise coherence governance are decision makers, influencers and enterprise architects. Design Science research is initiated by the identification and representation of challenging phenomena in the problem domain [3].

As shown in the left part of Figure 1, the challenge addressed in the GEA research programme is the improvement of enterprise coherence governance.

Guideline 3: Design Evaluation. The utility, quality, and efficacy of a design artefact must be rigourously demonstrated via well-executed evaluation methods. Evaluation of an artefact can be done using empirical and qualitative research methods such as observational, analytical, experimental, testing or descriptive-oriented methods [4].

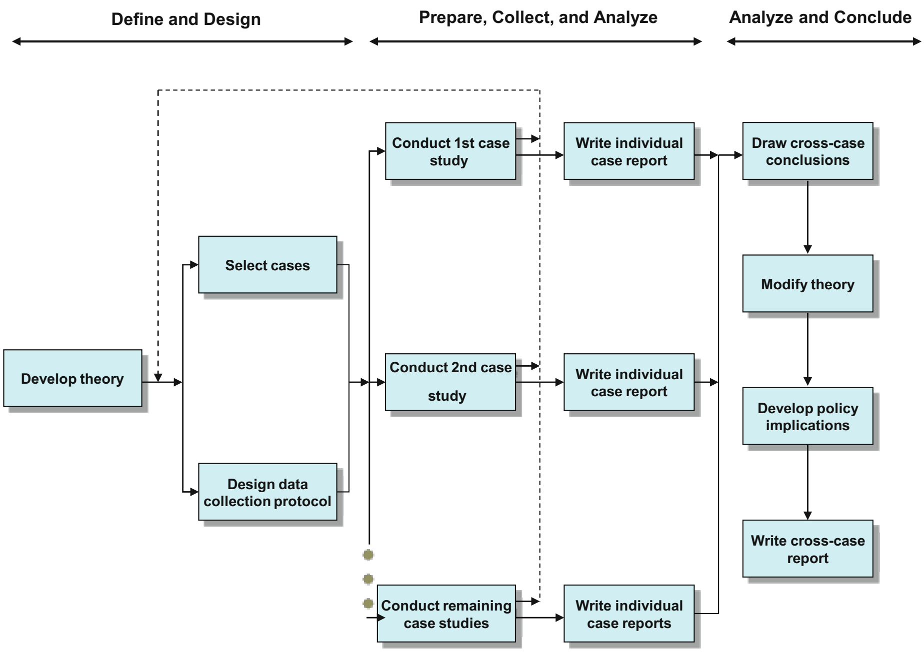


Fig. 2. Multiple case study research approach, adopted from Yin [32]

The lower box in the middle part of Figure 1 shows the evaluation method used in the programme, being the multiple case study research approach from Yin [32] as illustrated in Figure 2.

Guideline 4: Research Contributions. Effective Design Science research must provide clear and verifiable contributions in the areas of the design artefact, design foundations, and/or design methodologies [4].

The lower middle part of Figure 1 shows that the main contribution of the research programme to the knowledge base consist of the GEA theory and its artefacts ECA, ECF and ECG as well as the publications reporting on its effect in practise.

Guideline 5: Research Rigour. Design Science research relies upon the application of rigourous methods in both the construction and evaluation of the design artefact. Design Science artefacts are created based on existing foundations and methodologies in a knowledge base which include theories, frameworks, instruments, constructs, models, methods, instantiations, experiences, and expertise [4].

The right part of Figure 1 shows the foundations and methodologies (i.e. existing literature) that we adopted in the development of GEA. The bottom of the left part of Figure 3 shows that existing literature and experiences was be applied in the design, evaluation, and modifying phases of the research programme.

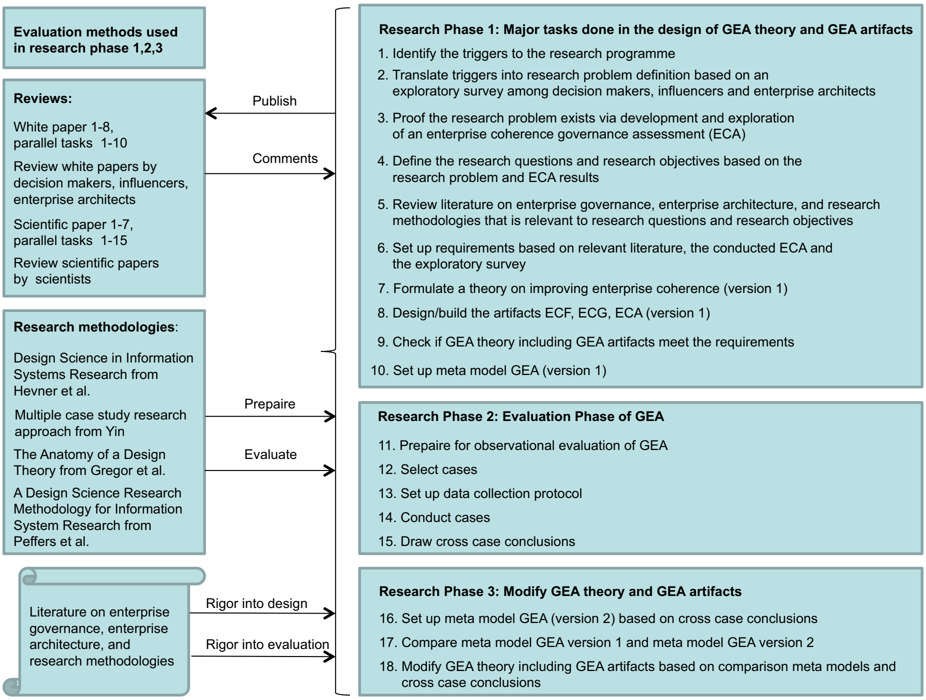


Fig. 3. Activities conducted to achieve the research objectives

Guideline 6: Design as a Search Process. The search for an effective artefact requires utilizing available means to reach desired ends while satisfying laws in the problem environment [4]. Design involves iterative research activities such as constructing, evaluating, and refining the artefact based on findings [3].

The major design activities to carry out in the research programme to achieve the research objectives are shown in the right part of Figure 3. It also shows that all the development activities to conduct in the research programme have been grouped into phase 1, phase 2 and phase 3. The result of phase 1 consist of the first version of GEA and the evaluation phase 2 formed the basis to modify GEA during phase 3 into the final version of GEA. In the left part of Figure 3 the evaluation methods that were used in the research programme are shown.

While using the Design Science approach from Hevner et al. [3, 4] as the overal research approach, the GEA programme also used:

- The *Multiple Case Study Research Approach* by Yin [32] to conduct phase 2.
- The *Anatomy of a Design Theory* by Gregor et al. [2] was used to evaluate GEA.
- The *Design Science Research Methodology for Information System Research* by Peffers et al. [7] was used as a 'template' for the GEA design process.
- In order to develop the initial versions of several of the GEA artefacts the group decision technique MetaPlan [8] was used.

Note: When the GEA programme was initiated in 2007, some of these approaches were not yet available [2, 7]. However, soon after their publication, these approaches were indeed integrated in the planning of the GEA programme as shown in Figure 3.

Guideline 7: Communication of Research. Design Science research must be presented effectively both to technology-oriented as well as management-oriented audiences. Communication of the research programme was targeted both at academic and industrial audiences, more specifically leading to several scientific publications [17, 18, 19, 20, 21, 22, 23, 24] and industrial publications [9, 10, 11, 12, 13, 14, 15, 16, 25, 26, 27, 28, 29, 30, 31]. The latter class of publications also played an important role in ensuring the relevance of the research results, as well as in ensuring a continued access to relevant case studies. This also made it necessary to target these publications at the local Dutch speaking EA community.

3.3 Research Activities of the GEA Programme

In fulfilling the above guidelines, the GEA programme followed the relevance, rigour, and design cycles as identified by Hevner et al [3], and in particular the design research process as suggested by Peffers et al. [7]. This process is illustrated in Figure 4.

The resulting process as followed by the GEA programme was already illustrated in Figure 3. The relevance, rigour, and design cycles that are typical of design science, are present throughout the listed research activities. For example, the relevance cycle is represented by tasks 2, 3, 11, 12, 13, 14, 15. The rigour cycle is represented by the adoption and application of existing scientific literature. The design cycle is represented in Figure 3 by the activities in tasks 1-4, 6-10 and 16-18.

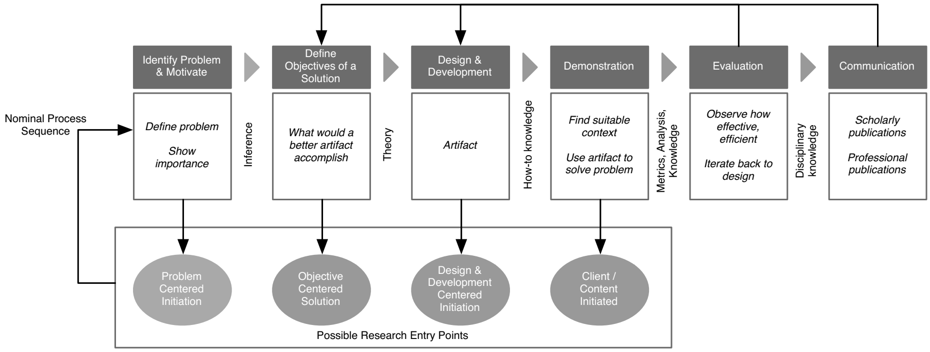


Fig. 4. Design research [7]

4 Implementation of the GEA Programme

4.1 Setup of the Research Programme

The GEA research programme itself was structured in terms of four groups:

1. A core team consisting of six to eight people with in-depth knowledge in the field of enterprise architecture.
2. A customer reference group representing the twenty major enterprises who *co-financed* the programme. These representatives involved policy makers, managers of enterprise architecture departments and lead enterprise architects.
3. An expert review team involving thirty lead enterprise architects.
4. A steering committee with seven leading representatives from science and industry.

To participate in one of these groups, candidate participants had to meet strict criteria:

1. For the core team, the participants were required to have in-depth knowledge of the field of enterprise architecture and a willingness to spent a considerable amount of time on the activities of the research programme.
2. For the customer reference group, participants were required:
   – to be working in one of the participating client organizations,
   – have a strong affinity with the discipline of enterprise architecture,
   – identify with the triggers and research problem of this research,
   – to be willing to discuss the interim results of this research on a regular basis and
   – to be willing (as an organization) to co-fund the research effort.
3. To be part of the expert review team, members were required to have an in-depth knowledge in the field of enterprise architecture and the willingness to thoroughly evaluate the interim results of this research within a relatively short time-span.
4. The scientific members of the steering committee were required to be among the leaders in their field. The industrial members of the steering committee were required to be employed/working at the boardroom level. All members of the steering committee were required to have the willingness (and commitment) to discuss the innovation strategy of this research on a regular basis.

The (potential) interests for the participants to be involved in this study were:

1. For the members of the core team it was an opportunity to follow their deep passion for the field of enterprise architecture, and be part of an effort to improve the field.
2. For the members of the customer reference group, participation provided a way to acquire better instruments to resolve enterprise coherence problems in practise.
3. For the members of the expert review team it was the opportunity to enrich their knowledge in the field of enterprise architecture with cutting edge insights.
4. For the scientific members of the steering committee it provided a way to obtain closer contact with relevant parties in industry. For the industrial members of the steering committee it provided a way to obtain better governance instruments.

The actual involvement (and composition) of these groups depended on the specific phase of the research programme. The results of the research activities, as conducted by the core team, were regularly discussed with (and evaluated by) the customer reference group. The members of the expert review team were charged with the responsibility to guard the quality by thoroughly evaluation the results. The steering committee regularly assessed the progress of the programme and the development strategy followed.

4.2 Development Process

As one of the first activities, the ECA (Enterprise Coherence-governance Assessment) instrument was developed and used in seven organizations to assess their level of enterprise coherence. The primary aim of the exercise was to assess if the perceived problem (*Many organizations lack enterprise coherence governance*) actually manifested itself in practise. Furthermore, jointly with the partners who co-invested in the research programme, a go/no-go criterion was formulated: *if more than 50% of the seven organizations involved in the first ECA studies lack enterprise coherence governance, it was safe to assume that the lack of enterprise coherence governance is indeed a relevant issue that needs further elaboration by the GEA programme*.

The secondary goal for the development of the ECA, and its application in seven participating organizations, was to gain an initial insight into research question number one and three: *What are the core factors that define enterprise coherence? What impacts does the governance of enterprise coherence have on the performance of enterprises in practice?* On the one hand, the answer to these questions provided insight into the need to carry out further research into the governance of enterprise coherence, while on the other hand providing a first refined definition of enterprise coherence and its practical impact on organizational performance.

At the start of the GEA research programme, the intention was to actually conduct three of such ECA studies during the programme:

1. A first assessment at the start of the programme, providing a baseline measure.
2. A second assessment once a shared understanding of enterprise coherence was reached. By comparing the results to the baseline, the effect of having a shared awareness of the forces that influence coherence should be measurable.
3. A final assessment once proper/full governance of enterprise coherence would be put in place in a participating organization. By comparing these final assessment

results to the earlier ones, the additional effect of coherence governance could be made explicit.

Nevertheless, soon after the start of the research programme, it became apparent that doing these three assessments was not feasible. In the time needed for such longitudinal assessments, the composition of the involved organizations, as well as the people involved, would change so much that the results would no longer be comparable between the first, second and third moment of measuring. It was therefore decided to integrate the ECA into the GEA method, and use the ECA in each of the cases in which GEA would be applied, and use its outcomes to drive the research activities in phase 2 (see Figure 3), in particular using Yin's [32] case based approach.

4.3 Case Studies

During phase 2 (see Figure 3), the GEA theory (or parts thereof) was applied in multiple organizations with the goal to evaluate the theory, and make improvements where needed. More specific goals for the case studies were: *Does the GEA theory meet its requirements?*, *Are the artefacts as defined by GEA producible in practise?*, and *Is GEA effective and applicable as a governance instrument?*.

In selecting the case studies, the focus was on large organizations involving a broad application of the GEA results. The primary case studies included:

- Professionalization in administrative body of a ministry of the Dutch goverment [23].
- An impact assessment of the introduction of a new law at the Custodial Institutions Agency of the Dutch Ministry of Public Safety and Justice [24].
- Digitization of the document flow at a ministry of the Dutch government [20, 21].

In line with Yin [32], the case studies involved the use of an explicit *data collection protocol*. Yin suggests to use five levels of questions in collecting data:

1. Questions to specific interviewees.
2. Questions at the level of an individual case (these are the questions in the case study protocol to be answered by the investigator during a single case, even when the single case is part of a larger, multiple-case study).
3. Questions focused on finding patterns across multiple cases.
4. Questions at the level of the entire research effort (for example, calling on information beyond the case study evidence and including literature or published data that may have been reviewed).
5. Normative questions about policy recommendations and conclusions, going beyond the narrow scope of the study.

The specific questions are, for each level, given below. These questions were validated by the core team of the GEA programme.

Level 1:

- At the time of the validation process of the ECF:
    1. Are the guiding statements valid and up to date?

   2. Do the representatives of the perspectives agree with the identified perspectives, the identified core concepts within it and the related guiding statements?
- At the time of the ECG analysis process of a major business issue:
   1. Do the causes, triggers, sub problems, risks, implications, etc. of the business issue lead to change initiatives?
   2. Do the (existing) guiding statements result in additional change initiatives or restrictions (the so called solution space)?

Level 2:

- Are the documents at the level of purpose present and accessible?
- Does the definition of the level or purpose result in a clear understanding of the sense of purpose and design of the organization?
- Is one able to identify, and engage, the right representatives for each of the perspectives? This engagement should cover both the identification and validation of the cohesive GEA elements and the analysis processes to solve the business issue.
- Are the representatives of the perspectives able to validate the ECF?
- Are the representatives of the perspectives, using the validated ECF, able to execute the analysis processes to solve major business issues?
- Does the development of the ECF lead to an increase of enterprise coherence?
- Does the use of GEA lead to an integral solution that contributes to the coherence of the organization?
- Is the organization able to, autonomously, specify a business issue that can serve as input to a GEA based analysis?
- Do the owners of the business issue succeed in specifying the business issue in such a way the representatives of the prospects can perform the complete GEA analysis and develop an integral solution?

Level 3:

- The level 3 questions about the pattern of findings across multiple cases are:
   1. Degree of acceptance by stakeholders?
   2. Extend of applicability?
   3. Extend of matching required dynamics?
   4. Extend of compliance with required integrality?
   5. Degree of accessibility?
   6. Degree of transferability?
   7. Extend of balance of interests?
   8. Degree of innovativeness?

Level 4:

- Did the execution of the cases result into detectable performance improvements?
- Does the literature support the answers to the above findings?

Level 5:

- What recommendations can be made towards the further development and expansion of the area of enterprise coherence?

The data gathering, structuring and analysis resulted in the validation of evaluation criteria such as: *degree of acceptance by stakeholders*, *extend of applicability*, *achieved level of coherence governance*, *degree of transferability*, *balance between interests of different stakeholders*, and *level of innovation*.

5 Results of the GEA Programme

The core result of the GEA programme is the GEA method, being a (situational) design theory on how to achieve enterprise coherence, involving three main components: ECA (Enterprise Coherence-governance Assessment) [17, 22], ECF (Enterprise Coherence Framework) [18] and ECG (Enterprise Coherence Governance-approach) [19, 23].

Scientific results of the GEA programme have been published in several papers [17, 18, 19, 20, 21, 22, 23, 24], while results have been communicated to practitioners as well in terms of white papers and industry oriented publications [9, 10, 11, 12, 13, 14, 15, 16, 25, 26, 27, 28, 29, 30, 31].

Collectively, the GEA method components cover the GEA concepts as depicted in Figure 5, where one concept builds on the other, leading to a coherent whole. All the promises of the EA-Vision, such as improving the coherence of the organization, should be achieved through the execution of EA-Processes. The execution of the EA-Processes results in EA-Products that will direct change programmes and via this the enterprise coherence. EA-People are needed to carry out the EA-Processes and to produce the EA-Products. EA-People needs, to execute the EA-Processes, sufficient allocation of means in terms of time, budgets and tools. EA-People and the execution of EA-Processes need to be governed by EA-Governance. And finally to store a maintainable formal description of the formulation of the EA-Vision, EA-Processes, EA-Products, EA-People and EA-Governance there is a need for an EA-Methodology. The ECG binds all these concepts together in a workable procedure for doing enterprise architecture.

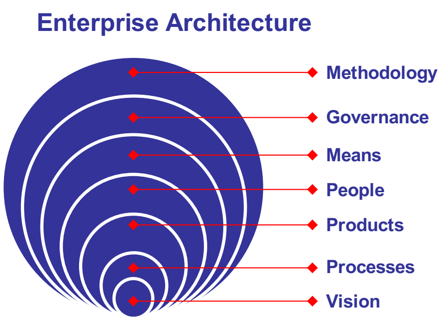


Fig. 5. Coherent set of GEA concepts

Using the criteria from Gregor et al. [2], the results can be summarized as follows.

Purpose and Scope. The purpose and scope of the GEA method is the development of an approach for understanding, governing and improving enterprise coherence

Constructs. The constructs the GEA method consists predominantly of the ECF, which makes an enterprise's coherence explicit, an ECG to govern and to improve the enterprise coherence, and an ECA to measure the level of enterprise coherence governance.

Principle of Form and Function. The GEA method is a methodology to aid in the identification of the level of enterprise coherence and issues a set of processes and heuristics to measure and improve the enterprise coherence governance.

Artefact Mutability. Suggestions for improving the GEA method are given for further work and research: one example is further research in applying the GEA method in cases of strategic alliances.

Testable Propositions. It is claimed that the GEA method is adaptable to situational settings, despite of its positioning as a general approach.

Justificatory Knowledge. The GEA method is derived from other governance theories and enterprise architecture approaches and based on several design theories.

Principles of Implementation. It is stated that implementation of the GEA method requires facilitation by a facilitator experienced in enterprise architecture, governance approaches, assessment approaches and running collaborative workshops.

Expository Instantiation. Three key examples of the application of the GEA method are given in large real world cases.

6 Conclusion

In this paper we discussed the research methodology used in the GEA (General Enterprise Architecting) research programme. The goal of this multi-client (involving twenty organizations) research programme was the development of a new approach for doing enterprise architecture. We discussed the motivations for starting the GEA programme, its focus, as well as the objectives. Based on this, the research methodology used by the GEA programme was discussed. This involved a combination of design science research and case study based research. We also elaborated on the way the GEA programme went about to actually implement the research methodology in a real-world situation, while also highlighting the results of the programme. The programme was structured around four groups of participants: a core team, a customer reference group, an expert review team and a steering committee. In the course of the programme, intermediate results were published. Not only targeted at an academic audience, put specifically also at an industrial audience to ensure continued support (in the market) for the GEA method during its development.

The most important lesson learned of applying the chosen research methodologies is the assurance of a high quality of the GEA method in terms of being relevant, rigour, proper evaluated and proper designed.

References

1. Lessen uit ICT-projecten bij de overheid, Deel B. De Algemene Rekenkamer (2008) (in Dutch)
2. Gregor, S., Jones, D.: The anatomy of a design theory. Journal of the Association for Information Systems 8(5), 312–335 (2007)
3. Hevner, A.R.: A Three Cycle View of Design Science Research. Scandinavian Journal of Information Systems 19(2), 87–92 (2007)
4. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28, 75–106 (2004)
5. Iivari, J.: A Paradigmatic Analysis of IS as a Design Science. Scandinavian Journal of Information Systems 19(2), 39–64 (2007)
6. March, S.T., Smith, G.F.: Design and natural science research on information technology. Decision Support Systems 15(4), 251–266 (1995)
7. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. Journal of Management Information Systems 24(3), 45–77 (2007)
8. Schnelle, E.: Neue Wege der Kommunikation. Spielregeln, Arbeitstechniken und Anwendungsfälle der Metaplan-Methode. Number Heft 10. Hanstein, Königstein/Taunus, Germany (1978) (in German)
9. Wagter, R.: Sturen op samenhang op basis van GEA – Permanent en event driven. Van Haren Publishing, Zaltbommel (2009) (in Dutch)
10. Wagter, R., Mahakema, M., Stovers, R., Krijgsman, W., Witte, D.: TOGAF en GEA: een gouden koppel? TIEM (44) (Maart 2012) (in Dutch)
11. Wagter, R., Nijkamp, G., Proper, H.A.: Applying GEA to a business issue. White Paper GEA-5, Ordina, Utrecht, The Netherlands (2007) (in Dutch)
12. Wagter, R., Nijkamp, G., Proper, H.A.: GEA-Benchmark: Measuring cohesion. White Paper GEA-3, Ordina, Utrecht, The Netherlands (2007) (in Dutch)
13. Wagter, R., Nijkamp, G., Proper, H.A.: Overview 1th Phase - General Enterprise Architecturing. White Paper GEA-1, Ordina, Utrecht, The Netherlands (2007) (in Dutch)
14. Wagter, R., Nijkamp, G., Proper, H.A.: The Elements of the "GEA-Structure". White Paper GEA-2, Ordina, Utrecht, The Netherlands (2007) (in Dutch)
15. Wagter, R., Nijkamp, G., Stovers, R., Proper, H.A.: E-Government using GEA and NORA. White Paper GEA-4, Ordina, Utrecht, The Netherlands (2007) (in Dutch)
16. Wagter, R., Nijkamp, G., Witte, D., Proper, H.A.: GEA in relation to other steering instruments. White Paper GEA-8, Ordina, Utrecht, The Netherlands (2008) (in Dutch)
17. Wagter, R., Proper, H.A., Witte, D.: Enterprise Coherence Assessment. In: Harmsen, F., Grahlmann, K., Proper, E. (eds.) PRET 2011. LNBIP, vol. 89, pp. 28–52. Springer, Heidelberg (2011)
18. Wagter, R., Proper, H.A., Witte, D.: Practice-Based Framework for Enterprise Coherence. In: Proper, E., Gaaloul, K., Harmsen, F., Wrycza, S. (eds.) PRET 2012. LNBIP, vol. 120, pp. 77–95. Springer, Heidelberg (2012)
19. Wagter, R., Proper, H.A., Witte, D.: Enterprise Architecture: A strategic specialism. In: Rinderle-Ma, S., Sanz, J., Bai, X.-Y. (eds.) Proceedings of the 14th IEEE Conference on Commerce and Enterprise Computing (CEC 2012), pp. 1–8. IEEE Computer Society, Hangzhou (2012)
20. Wagter, R., Proper, H.A., Witte, D.: Enterprise Coherence in the Dutch Ministry of Social Affairs and Employment. In: Bajec, M., Eder, J. (eds.) CAiSE 2012 Workshops. LNBIP, vol. 112, pp. 600–607. Springer, Heidelberg (2012)

21. Wagter, R., Proper, H.A., Witte, D.: On the Use of GEA at the Dutch Ministry of Social Affairs and Employment. In: Rinderle-Ma, S., Sanz, J., Bai, X.-Y. (eds.) Proceedings of the 14th IEEE Conference on Commerce and Enterprise Computing (CEC 2012), pp. 115–119. IEEE Computer Society, Hangzhou (2012)
22. Wagter, R., Proper, H.A., Witte, D.: The Extended Enterprise Coherence-governance Assessment. In: Aier, S., Ekstedt, M., Matthes, F., Proper, E., Sanz, J.L. (eds.) PRET 2012 and TEAR 2012. LNBIP, vol. 131, pp. 218–235. Springer, Heidelberg (2012)
23. Wagter, R., Proper, H.A., Witte, D.: A theory for Enterprise Coherence Governance. In: Saha, P. (ed.) A Systemic Perspective to Managing Complexity with EA. IGI Publishing (to appear, 2013)
24. Wagter, R., Proper, H.A., Witte, D.: Enterprise Coherence Governance in the Public Sector. In: Submitted to CBI 2013, 15th IEEE Conference on Business Informatics (2013)
25. Wagter, R., Stovers, R., Nijkamp, G., Proper, H.A.: GEA-processes and products, a closer examination. White Paper GEA-6, Ordina, Utrecht, The Netherlands (2007) (in Dutch)
26. Wagter, R., Witte, D.: GEA in relation to other steering instruments. White Paper GEA-8 (May 2008) (in Dutch)
27. Wagter, R., Witte, D.: Strategische stuurinstrumenten vergeleken. TIEM (33) (October 2009) (in Dutch)
28. Wagter, R., Witte, D.: Sturen op samenhang op basis van GEA. XR-magazine (May 2010) (in Dutch)
29. Wagter, R., Witte, D.: Toepassing van het strategische stuurinstrument GEA. TIEM (35) (February 2010) (in Dutch)
30. Wagter, R., Witte, D., Goutier, H., Lieshout, J.: NORA 3.0 als stuurinstrument op basis van GEA. TIEM (39) (November 2010) (in Dutch)
31. Wagter, R., Witte, D., Proper, H.A.: The GEA architecture function: A strategic specialism. White Paper GEA-7, Ordina, Utrecht, The Netherlands (2007) (in Dutch)
32. Yin, R.K.: Case Study Research – Design and Methods, 4th edn. Sage Publications (2009)

Towards an Engineering-Based Research Approach for Enterprise Architecture: Lessons Learned from Normalized Systems Theory

Philip Huysmans and Jan Verelst

Normalized Systems Institute, University of Antwerp, Antwerp, Belgium
{philip.huysmans,jan.verelst}@ua.ac.be

Abstract. The emerging field of enterprise engineering provides a promising outlook for positioning relevant research. Enterprise Architecture frameworks which are frequently used in practice, but are often criticized from a research perspective, can be positioned in this field. The challenge for the enterprise engineering field is to provide a framework to improve such frameworks using a rigorous scientific approach. This paper aims to contribute to addressing this challenge by proposing components for a research framework which focuses on applying engineering insights to enterprise architecture. It first explores how current enterprise architecture frameworks handle issues relevant for engineering (i.e., complexity, change and integration). It then introduces additional components which could contribute towards a more systematic approach. These components are derived from the way the Normalized Systems Theory was developed, and successfully introduced engineering standards into the design software architecture.

Keywords: Enterprise Architecture, Enterprise Engineering, Normalized Systems.

1 Introduction

Enterprise engineering is an emerging field which can benefit from many research approaches. One approach is to consider frameworks, methods and analysis techniques in the context of traditional engineering sciences. While certain issues faced in an organizational context will not be addressed by this approach, it limits the scope of issues to a more systematic analysis. Examples of typical issues which are addressed by engineering sciences are complexity, change and integration. These issues are very relevant for enterprises as well.

Integration is an often-recurring theme in many organizational research domains. It refers to the need of an organization to operate as a unified whole, which can be guided in a strategic direction (cf. management research) or is recognized as a distinct brand by customers (cf. marketing research). Within the information systems research domain, the integration issue becomes most apparent in Business/IT alignment research [1]. Even without considering supporting IT functions, organizations need to be able to integrate their product portfolio,

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 58–72, 2013.

the production processes used to create these products, and the organizational structures which execute these processes. These different components are generally considered to increase in *complexity* over time. First, more sophisticated products are needed to avoid a competition based solely on price, while leads for example to the servitization of products (i.e., bundling products and services) [2]. Second, the demand for increased quality (e.g., TQM) and responsiveness (e.g., JIT) requires more complex production processes. Third, globalization and frequent mergers and acquisitions often result in more complex organizational structures, or require a complex reorganization to obtain a simpler structure. Fourth, IT systems has been described to become increasingly complex, as described by Lehman's laws [3]. Additionally, organizations need to be able to respond quickly to *changes* in their environment in order to maintain a competitive advantage [4,5]. Moreover, changes cannot be considered to be individual or isolated. Rather, it is claimed that different changes need to be applied at a steady pace in order to adequately react to contemporary markets [4,5]. This means that it is difficult to describe a stable state of an enterprise to use as a starting point for change, and that different change projects can interact with each other.

Within the enterprise engineering field, enterprise architecture frameworks can be considered to contain the most advanced knowledge to address these issues. The ANSI/IEEE 1471-2000 standard defines architecture as "the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution" [6]. This definition distinguishes between descriptive and prescriptive perspectives on enterprise architectures. In a descriptive perspective, the representation of the structure of elements and the relationship between them is the main contribution of an architecture. This perspective is usually associated with a blueprint of an organization as a system. In a prescriptive perspective, the design of organizational artifacts is guided by specifying principles or regulations which limit the design freedom for individual artifacts. By adhering to these principles, a well integrated system will be designed. Various enterprise architecture frameworks are explicitly descriptive (e.g., "enterprise architecture is a set of descriptive representations of an enterprise" [7]) or prescriptive (e.g., "enterprise architecture is a coherent whole of principles, methods, and models that are used in the design and realization of an enterprise's organizational structure, business processes, information systems and infrastructure." [8]).

Both types of enterprise architecture framework propose to deal with the identified issues of complexity, change and integration in different ways. Certain authors claim that, because of the radical differences between two types of approaches, they are complementary instead of comparable (e.g., [9]). In this paper, we explore how current approaches deal with the issues of integration, complexity and change, and provide an outlook on future research directions to works towards a more deterministic approach. Notwithstanding their differences, similarities between enterprise architecture frameworks can be observed. Therefore, we first explore some general characteristics of enterprise architecture frameworks in Section 2. Next, we describe more in-depth how the identified issues are addressed in two mainstream enterprise architecture frameworks in Section 3, in order to describe the current state of the field more in-depth. We then discuss in Section 4 how current research initiatives attempt to advance the enterprise engineering field by explicitly applying engineering concepts to software, process and organizational domains. Following this discussion, we propose research directions which are more closely related to a traditional engineering approach to address current research gaps in Section 5.

2 How Complexity, Integration and Change Are Handled in Enterprise Architectures

2.1 Reducing Complexity by Applying Abstraction

Enterprise Architectures comprise a large amount of concepts, entities and variables, which cannot be retained or managed at once. Frameworks aim to reduce complexity by creating abstractions from real-world artifacts by creating models [10]. According to Bernus et al. [8], complexity can be measured using a function of the number of elements and relations in a system. Following this definition, complexity can indeed be reduced by omitting classes of elements and relations. As a result, smaller and simpler models can be created, which focus explicitly on a certain aspect and neglect other aspects. Various abstraction mechanisms, such as perspectives, viewpoints, areas of concern, dimensions, etc. are used to limit the scope of a certain class of models or principles [11]. This approach can aid understandability and enable easier communication between stakeholders in the organization, which is indeed considered to be an important advantage of descriptive enterprise architecture frameworks. However, the application or aspiration of enterprise architecture frameworks is not limited to communication. Enterprise architecture frameworks (especially prescriptive ones) are applied to *design* organizations and supporting systems as well [8,12]. During design, certain aspects of a system cannot be abstracted without considering their impact. Certain authors therefore argue that enterprise architecture design principles should be augmented with standards and guidelines [13]. In this way, each architectural layer can be designed by adhering to a specific set of guidelines.

2.2 Achieving Integration

Enterprise architecture frameworks originated based on the need for integration between the strategic orientation of an organization and its concrete operation. The identification of the need for such an integrated view can be traced back to organizational theory [14]. It was observed that the formal structures, goal orientations and time orientations between sales, research, and production divisions in large industrial organizations were very different from each other. A higher degree of differentiation between divisions was found to be inversely related to the degree of integration of the organization. While the differentiation of the divisions was explained as a reaction to the variety of requirements from their environment, the effective performance of the organization requires the integration of the outcomes of every division [14]. Various authors have identified organization-wide communication and decision-making as crucial preconditions for this integration [15,16]. Consequently, any organization-wide approach such as the specification of a strategic plan or an architecture can be considered as a contribution to integration and alignment [17].

However, when integration is considered to be more specific (as in: integration between architectural layers), contributions are less frequent. Most frameworks focus on the specification of the different layers which need to be defined, and modeling of the separate layers. As discussed above, this is mainly aimed at the reduction of complexity. Less focus is on the integration between layers. Most publications on enterprise architecture research therefore report on contributions which can be located on a single layer, while few authors address integrating multiple layers [18]. The capability to deal with the relations between model elements on various levels of the enterprise architecture is also referred to as traceability. Wegmann considers this traceability to be essential for enterprise architecture, as it makes the integration between different levels explicit. However, he acknowledges that, while enterprise architecture frameworks should be created to enable this traceability, it is difficult to clearly establish and maintain between the levels [19].

2.3 Dealing with Change

A specific distinction between the handling of change can be observed between descriptive and prescriptive enterprise architecture frameworks. Descriptive frameworks consider an organization as a static system, at a given moment in time. The models created for this static system are referred to as as-is models. When changes need to be introduced, a new version of the framework (i.e., a to-be version) is created, which specifies a future static state. Through gap analysis, change projects can be defined which guide the organization from its as-is to to-be state. However, a method to implement these changes is often out of scope for these frameworks [7].

Prescriptive frameworks, on the other hand, claim to focus more explicitly on providing guidance for change. According to the ANSI/IEEE STD 1470-2000 definition, principles should be defined to govern both the design and *evolution* of enterprise architecture. Other authors consider the very nature of architecture (and hence, principles) to be the *limitation of design freedom* [20]. However, many of the frameworks suggest to create company-specific principles, without providing a way to evaluate the applicability of such principles in other contexts. This limits the generalizability of these principles, and obstructs their systematic application. Nevertheless, this is exactly what is expected from a mature enterprise engineering field.

3 Exploring Specific Enterprise Architecture Frameworks

In this section, we explore whether the Zachman and TOGAF enterprise architecture frameworks provide systematic guidelines to deal with complexity, integration and change.

3.1 Zachman

The Zachman framework [21,7] is an enterprise architecture framework which is often referenced amongst practitioners and researchers. According to several authors, the Zachman framework has the widest adoption in enterprise architecture projects [22,23]. Moreover, it is often used as a basis for evaluating, establishing and customizing other frameworks [23]. Some other frameworks literally position themselves as extensions based on the Zachman framework (e.g., FEAF [24]). Therefore, it is important to understand how the Zachman framework is originally developed. Otherwise, assumptions can be made based on which incorrect solutions for the identified enterprise architecture issues can be suggested. The Zachman framework deals with change and complexity as described in Section 2 for descriptive frameworks: it uses abstraction in different perspectives and dimensions to reduce complexity, and proposes to develop different versions of the framework to deal with changes. In order to achieve integration between models in different cells, different mechanisms are described: constraints, dependencies, and model transformations.

Constraints: The models from each perspective (i.e., each row) have a different set of constraints they need to adhere to [7]. For example, the models in the scope row are subjected to usability constraints (e.g., utility of the artifact), while models in the technology row are subjected to constraints from the state of the art of the used implementation platform. These constraints are additive across the different perspectives: constraints of a lower row also limit the models from higher rows. For example, the technological constraints on the technology models (e.g., only webservices are allowed) will also impact the system model, which will need to structure the system using services. When a constraint in a lower row is inconsistent with a model defined in a higher row, "the designers who are responsible for the two rows must initiate a dialog to determine what must be changed and to ensure that no gap in expectations exists between the different perspectives" [7]. Consequently, it is argued that the issue of conflicting constraints can be handled by communication.

Dependencies: It is acknowledged that the different cells describe abstraction of the same underlying organization, and that dependencies between the cells have an impact. At a minimum, the cells are considered to be related to every other cell in the same row [7]. If a change in the structure of one cell affects the structure of another cell, a dependency between these cells exists. Moreover, such dependencies can occur not only within a row, but also between rows.

The framework however does not provide guidance to determining the possible impacts between models from different cells. It is however acknowledged that the challenge during designing a model "is to design each while understanding the integrity of all others to avoid being surprised by undesirable side effects appearing long after it is possible to contain them" [7, p.595], and that understanding and storing the dependencies would "constitute a very powerful capability for understanding the total impact of a change" [7, p.603].

Model Transformations: While the idea of building IT systems using the Zachman framework is mainly claimed by other authors, Sowa and Zachman already mention the idea of *model transformations* [7]. For example, Pereira and Sousa elaborate on the notion of model transformations by specifying which models are used as input for certain cells [25].

While these mechanisms acknowledge the need for integration, they lack concrete guidelines to achieve such integration. Rather, they either suggest ad-hoc solutions (i.e., handling constraints through communication), admit that no solutions exist (cf. dependencies), or provide no detailed discussion (cf. model transformation). Consequently, the Zachman framework can be considered as a purely descriptive framework, which does not aim to provide prescriptive guidelines.

3.2 TOGAF

Contrary to the Zachman framework, TOGAF [12] explicitly proposes to define principles to guide the development of different architectures. However, these principles are considered to be organization-specific. The provided principles are considered to be examples, which makes them less relevant to evaluate in a broader context. TOGAF is based on four architecture domains: business architecture, applications architecture, data architecture and technical architecture. Instead of focusing on the end products of the architecture, TOGAF focuses on the process to develop the different architectures. In order to develop these architectures, TOGAF suggests an Architecture Development Method (ADM). The ADM is usually considered to be the most important component in TOGAF. Therefore, TOGAF is generally considered to be a process-oriented framework, instead of a product-oriented framework such as Zachman.

TOGAF does specify the "architectural input" which are required for the different phases in the ADM. However, given the absence of a clear approach for integrating different architectures, no concrete way of working can be described. For example, applications and data need to be mapped to physical technological artifacts in phase D. A detailed step in the method description is then: "12.4.6: Resolve Impacts Across the Architecture Landscape" [12]. However, no further details are provided on how to resolve these impacts. Therefore, it is not clear how, for example, conflicting principles need to be resolved or dealt with.

TOGAF explicitly aims to be a starting point for developing an enterprise architecture, but needs to be extended with other methods. For example, it proposes to use Service Oriented Architecture-related methods to design the IT architecture. Other researchers have reported on research efforts towards this goal as well [26]. Nevertheless, TOGAF itself provides little guidance for the actual design of enterprise architecture artifacts.

4 Suggestions for Prescriptive EA Components

A current lack in enterprise architecture frameworks is the availability of prescriptive guidelines for designing organizational artifacts. Descriptive frameworks do not focus on providing prescriptive guidelines, and prescriptive frameworks either suggest organization-specific guidelines, or focus on defining a process to design architectures, without elaborating on the actual designed artifacts. In order to address this lack, we propose several components in this section which are based on the research initiatives related to Normalized Systems [27,28,29]. The theory on Normalized Systems implies a highly structured approach to software architectures, based on fine-grained modular structures and the systematic reuse of building blocks called elements. The resulting software architectures exhibit behavior that we consider quite different from less structured approaches. More specifically, the result can be considered more *deterministic*. Determinism is related to evolvability (i.e., impacts are easily identifiable as all applications share the same fine-grained modular structure), reuse (i.e., all applications consist of instantiations of the same elements), correctness (i.e., because all applications are based on the same elements, every bug only needs to be corrected once), and reliability and performance (i.e., the investment optimizing an element is economically feasible due to the systematic reuse of the elements). Consider for example how application development is made more deterministic: first, it requires the elements to be built in a certain technology environment, which requires highly advanced expertise in design of software architecture. Next, however, is a phase that is of remarkably lower complexity or expertise levels: building applications. It suffices to identify the required instances of elements, and performing the required manual coding. There is no architectural design phase in building these applications, as the design is already incorporated in the elements. This makes the development process quite similar for all applications, leading to increasing reuse of knowledge about the elements, applications and development process. In this way, Normalized Systems Theory has contributed to resolving wicked issues in software engineering regarding complexity, change and integration. As such, it can be considered a useful candidate to explore for dealing with these issues in the enterprise architecture field.

The components presented here had an essential role in developing the Normalized Systems theory, but are not only applicable to software architecture. They have emerged based on more generally applicable engineering knowledge [27], not on specific software architecture knowledge. Moreover, they have already been applied to the fields of business process design [30] and enterprise architecture [31]. Therefore, we suggest the use of these components as a basis for a research framework for enterprise engineering.

4.1 Modularity Guidelines

Even though originating from systems theory the modularity concept has caught the attention of engineers, managers and researchers in a large variety of fields [32]. Modularity is defined as a property of a complex system, whereby the system is decomposed into several subsystems (i.e., modules). Obviously, each of these modules ultimately must cooperate with other modules in order to ensure the adequate functionality of the system as a whole. The interaction of a module with its external environment should be exhaustively and unambiguously documented in its interface. The interface describes the inputs required by the module to perform its part of the functionality, and the output it will provide to its external environment. As soon as such an interface is designed, one may learn about the intermodular dependencies, i.e., what does a module require from the other modules to perform its own functionality and what is the impact of a change in the module design for other modules.

From a modularity perspective good modular design is characterized by: (a) low intermodular coupling (i.e., few intermodular dependencies), and (b) high intramodular cohesion (i.e., strongly related and dependent elements within a module). From a practical point of view good modular design implies that: (a) changes in the design of one module have no or only a limited impact on the design of other modules, and (b) the function of one module can be studied in more or less isolation from the rest of the system. Consequently, a well-designed modular system enhances the comprehensibility and decreases the complexity of the overall system.

Enterprise architectures can be considered to be modular structures [33,34,10]. On each architectural layer, artefacts are defined which provide functionality required on higher layers, and use functionality from lower layers. Their prescriptive design should then be guided by modularity principles. Modularity has already been proposed by several authors to guide enterprise architectural design [33,34]. Unfortunately, good modular design is far from trivial because many architectural decisions have to be taken. Therefore, more in-depth guidelines are needed which demonstrate how modularity guidelines should be applied in an enterprise architecture context. Hence, a theory which prescribes principles to guide the design of a good modular design is highly desirable. For instance, Normalized Systems theory relies on modularity principles to define more specific software-related principles for achieving an evolvable software structure.

4.2 Preventing Combinatorial Effects

When more insight is gained in the modular structure of enterprise architecture, more specific concepts which build on modularity can be introduced. For example, *combinatorial effects* have been identified as the main obstacle for achieving evolvability in software architectures in Normalized Systems Theory. A combinatorial effect occurs when the effort required to apply a certain change increases with the size of the system. While combinatorial effects have initially been identified at the software level, subsequent research has identified them at the business process level and enterprise architecture level as well [30,31]. At the business process level, the modular structure has been described as consisting of tasks and their compositions (i.e., process flows). Combinatorial effects have been identified at the level of both the task and process flow levels, and a set of guidelines has been formulated to prevent such combinatorial effects [30]. Consequently, these guidelines can be evaluated and elaborated upon. The resulting set of guidelines can be used to deterministically design business processes without combinatorial effects in any context. Similar to Normalized Systems Theory on the software level, the design can be called deterministic since it needs to adhere to guidelines which have been shown to introduce combinatorial effects when violated, without any assumptions regarding the specific organizational context in which they are applied. At the enterprise architecture level, modular structures have been identified as well, and combinatorial effects have been described [31]. These combinatorial effects have been grouped in two categories. A first category describes so-called *horizontal* combinatorial effects, which affect artifacts within a single enterprise architecture layer. Combinatorial effects at the software level or business process level are examples of such horizontal combinatorial effects. A second category describes *vertical* combinatorial effects. Vertical combinatorial effects can be caused on any layer, but have an impact on other layers. For example, design decisions at the application level have been observed to impact the organizational level [31]. Such combinatorial effect show why the abstraction in current prescriptive enterprise architectures frameworks may be insufficient. As discussed above, prescriptive enterprise architecture frameworks apply abstraction by defining principles on different layers. This implies that issues on each layer can be addressed by principles on that layer. This approach cannot adequately deal with vertical combinatorial effects. As a result, complementary efforts are required.

4.3 Functional—Constructive Gaps

General Systems Theory (GST) studies the general behavior and characteristics of systems. The functional and constructional perspectives on a system are fundamentally different conceptualizations of a system [35]. The functional perspective is concerned with the external behavior of the system [36]. This perspective is adequate for the purpose of using or controlling a system. Consequently, the actual construction of the system is not relevant. Instead, the focus is on how this system interacts with its environment. Therefore, knowledge of the required input variables, transfer function and output variables are key components of this perspective. The input and output variables represent which interactions occur with the environment. The transfer function describes how the input variables are transferred into output variables. This transfer function can be adjusted by control variables, which can alter the behavior of the system. This perspective uses so-called black box models to represent a system.

In contrast, the constructional perspective describes what a system really is [37,38]. In this perspective, knowledge about the composition (i.e., which components constitute the system) and structure (i.e., how these components are related) is important. Consequently, a fundamentally different kind of model is needed to represent this perspective. This type of model is called a white box model. Such models represent the construction and operation of a system. Analogously to functional decomposition, constructional decomposition can be applied to study the subsystems of a complex system.

Both perspectives need to be integrated: a functional requirement needs to be brought about by a constructional design. However, this relation is not straightforward. Therefore, a so-called *functional—constructive gap* exists, which needs to be bridged by a certain design. Prescribing such designs has proven to be extremely difficult. Nevertheless, further insight in such approaches is valuable, since an enterprise architecture could be considered as a series of functional—constructive gaps. On each layer of an enterprise architecture, different models are created in both a functional and constructive perspective. Insight in how models from both perspectives can be related to each would provide a better way to approach the integration issue in enterprise architectures. Normalized Systems theory has previously been described as a transformation from elementary functional requirements to a constructive software design [28].

4.4 Developing Domain-Specific Patterns

Certain authors argue that a domain-specific approach is required for the specification of prescriptive guidelines, because substantial differences exist between various domains. Such principles can be published in the form of patterns, which describe generic solution for a class of domain problems. This can be considered to be a middle ground between the stance that enterprise architecture principles can only be organization-specific, as discussed above, and that generally applicable principles exist, but have not yet been specified sufficiently. A domain-specific approach is increasingly feasible because business processes are currently being standardized in several sectors. Previously, organizations have developed their own business processes for a very diverse range of domains, including human resources, accounting, finance, order management, logistics and production. The software supporting these business processes was therefore also relatively different between organizations. Many custom-built systems were produced, and software packages needed extensive customizations to make them fit for the organization's processes. These idiosyncratic business processes made it expensive to develop sufficiently generic and flexible building blocks at the level of granularity of services because the cost could not be spread over multiple organizations and/or projects. It has been noted that "domain analysis is invariably conducted within one organization so that transfer of components between domains is difficult" [39]. However, recently a number of indications that organizations are starting to standardize their processes are occurring. Such indications are, amongst others, (a) increased adoption of enterprise software packages such as ERP systems; (b) initiatives to develop reference models of business processes; and (c) standards for communicating business information (ontologies). Domain-specific patterns could make a thorough, engineering-based evaluation feasible, since they are concrete enough to show design alternatives, and abstract enough to be applied in different organizational contexts.

5 Towards an Enterprise Engineering Research Framework

In this paper, we argued how typical engineering problems such as integration, change and complexity are relevant to organizations as well. Within the field of enterprise engineering, we explored how enterprise architecture frameworks suggest to deal with these issues. We differentiated between descriptive and prescriptive frameworks, and their different approaches to these issues. The current state of the art suggests that many descriptive product-based frameworks are available, together with several prescriptive process-based frameworks. While these frameworks can certainly aid the design of enterprise architectures in practice, they are insufficient for the scientific development of the enterprise engineering field. Instead, an increased focus on prescriptive, product-based components is required.

In the previous section, we discussed several components which could be constituents of a research framework for developing such prescriptive, product-based guidelines. Such a framework would consider an enterprise architecture as a series of functional—constructive gaps, of which each gap needs to be addressed adequately. In order to evaluate the transformation from a functional to a constructive perspective, a theory-based evaluation needs to be conducted. Combinatorial effects, which are based on stability and entropy, are proposed as a candidate for such an evaluation criteria. On the software level, the elimination of combinatorial effects has resulted in software architectures which are evolvable. It is important to note that the adequate identification of combinatorial effects needs to be done on the lowest, most detailed level of software design. Compare this to the tendency of enterprise architecture frameworks to promote high-level overview models. Such models do not contain enough detail to judge whether certain modularity principles are violated. This implies that such high-level overviews may be highly useful in attempts to comprehend an enterprise, but they are not sufficient in order to design an enterprise without combinatorial effects. Nevertheless, many of these enterprise architecture frameworks do make claims that they improve evolvability. This may still be true for example because comprehension is improved, but not from the point of view of a deterministic way to design architectures.

The identification of combinatorial effects requires an explicit statement of how the modular structure of the level which is addressed. Describing the structure of a certain domain requires a constructive perspective. Consequently, a bottom-up, domain-specific approach for developing enterprise architecture patterns should be aimed for. Through iterative refinement of such patterns, reusable building blocks can be created. These reusable building blocks should address all relevant concerns of their level. On the software level, Normalized Systems has shown that this is a requirement for achieving anthropomorphism, as advocated by the Object-Oriented Paradigm. Anthropomorphism can only be achieved when a building block can be used in a functional, black-box way, without the need to deal with concerns which are related to the construction of that building block.

This research approach suggests that no overarching enterprise architecture principles should be defined up front, since the complexity which is in scope is very large. Rather, a clear positioning of developed artifacts and the way in which they are evaluated allows a consistent maturing of the field. More specifically, this allows for various research initiatives, both in the short and long term. In the short term, such research goals could focus on a further specification of the research framework. First, a more specific scoping of the domains which can be designed and the issues which can be handled by design should be clarified. This could be related to, for example, research related to the identification of modular structures. While much research is already available on the modular structure of IT, products, business processes and organizational structures, other, less obvious, examples of modular structures are emerging, such as in the legal domain [40]. Methodologically, such research could benefit from an explorative case study approach. In order to observe relevant modular structures, in-depth knowledge of the field is required, which assumes a qualitative research approach. Such an approach allows the researcher to study a phenomenon which is not clearly separated from its environment [41]. This is necessary to observe modular dependencies which do not surface within the immediate environment of the phenomenon. Second, a specification of the functional—constructive gaps in an organizational context should be researched. This could related to, for example, architectural layers in enterprise architecture frameworks. Methodologically, a similar research approach as for the previous research goal could be used. However, researchers which actually implement enterprise architecture frameworks in organizations could provide valuable insights for this research goals. Therefore, an action research approach could be attempted as well. Third, a taxonomy of combinatorial effects should be created, which allows classification of empirical observations of such effects. Such a taxonomy could elaborate on the classification of horizontal versus vertical combinatorial effects [31]. This research goal could aggregate the research results of cases performed for the two previous research goals. Fourth, a taxonomy for different types of solutions could be proposed. On the software level, combinatorial effects are prevented by using pattern expansion. However, other solution types can be better suited for organizational designs, such as centralization or bus patterns. These solution types could build on generally applicable engineering knowledge. When exemplary solutions are observed in practice, these could be reported through explanatory case studies. However, the development of new solutions could be presented as genuine design science research.

On the long term, more mature research results can be aimed for. For specific domains, reusable patterns should be researched, which have a well-defined set of functionalities and handle a specific set of concerns. In this way, a more deterministic approach of building enterprise architectures can be achieved. For example, a pattern could be designed for an insurance or a production order. These patterns should be designed to deal with changes from not only their functional domain (e.g., procurement), but from other domain as well (e.g., legal concerns). Given the current state of the field, a design science approach with multiple iterations seems necessary. An important catalyst for this type of research would be an outlet where concrete artifact designs can be discussed and published.

6 Conclusions and Limitations

In this paper, we examined how enterprise architecture frameworks may improve the emerging field of enterprise engineering. We therefore examined how such frameworks deal with issues such as integration, change and complexity. We then proposed several components to work towards a research framework for performing enterprise engineering research, based on the approach which was used to develop the Normalized Systems theory. This theory has shown that prescriptive, product-based guidelines can be developed to approach software development adhering to engineering standards. We believe that a similar goal should be aimed for by the enterprise engineering community. Of course, not all enterprise architecture issues can be addressed by this approach. For example, social issues during enterprise architecture projects will not be resolved. The scoping of the issues which are relevant for an engineering approach therefore indicate the limitations of such an approach. While such issues are not denied, the issues identified in this paper can be considered to be relevant and unresolved. Focussing on such issues can therefore result in significant contributions for the enterprise architecture field.

References

1. Henderson, J.C., Venkatraman, N.: Strategic alignment: leveraging information technology for transforming organizations. IBM Syst. J. 32, 4–16 (1993)
2. Anderson, J.C., Narks, J.A.: Capturing the value of supplementary services. Harvard Business Review 73(1), 75–83 (1995)
3. Lehman, M.: Programs, life cycles, and laws of software evolution. Proceedings of the IEEE 68, 1060–1076 (1980)
4. Eisenhardt, K.M., Martin, J.A.: Dynamic capabilities: What are they? Strategic Management Journal 21(10/11), 1105–1121 (2000)
5. Teece, D.J., Pisano, G., Shuen, A.: Dynamic capabilities and strategic management. Strategic Management Journal 18(7), 509–533 (1997)
6. Maier, M.W., Emery, D., Hilliard, R.: Ansi/ieee 1471 and systems engineering. Systems Engineering 7(3), 257–270 (2004)
7. Sowa, J.F., Zachman, J.A.: Extending and formalizing the framework for information systems architecture. IBM Systems Journal 31(3), 590–616 (1992)
8. Bernus, P., Nemes, L., Smidth, G.: Handbook on Enterprise Architecture. In: International Handbooks on Information Systems. Springer (2003)
9. Martin, R., Robertson, E.: A comparison of frameworks for enterprise architecture modeling. In: Song, I.-Y., Liddle, S.W., Ling, T.-W., Scheuermann, P. (eds.) ER 2003. LNCS, vol. 2813, pp. 562–564. Springer, Heidelberg (2003)

10. Ross, J.W., Weill, P., Robertson, D.C.: Enterprise Architecture as Strategy – Creating a Foundation for Business Execution. Harvard Business School Press, Boston (2006)
11. Zarvic, N., Wieringa, R.J.: An integrated enterprise architecture framework for business-it alignment. In: Latour, T., Petit, M. (eds.) Proceedings of Workshops and Doctoral Consortium of the 18th International Conference on Advanced Information Systems Engineering (CAiSE 2006), pp. 262–270. Namur University Press, Luxembourg (2006)
12. The Open Group: The open group architecture framework (togaf) version 9 (2009), `http://www.opengroup.org/togaf/`
13. Korhonen, J.J., Hiekkanen, K., Lähteenmäki, J.: Ea and it governance — a systemic approach. In: Proceedings of the European Conference on Management, Leadership and Governance (ECMLG) (2009)
14. Lawrence, P.R., Lorsch, J.W.: Differentiation and integration in complex organizations. Administrative Science Quarterly 12(1), 1–47 (1967)
15. Hicks, H.G., Gullet, C.R.: Organizations: Theory and Behaviors. McGraw-Hill, New York (1975)
16. Gortner, H., Mahler, J., Nicholson, J.: Organization Theory: A Public Perspective. Dorsey Press (1987)
17. Gregor, S., Hart, D., Martin, N.: Enterprise architectures: enablers of business strategy and is/it alignment in government. Information Technology & People 20(2), 96–120 (2007)
18. Schöenherr, M.: Towards a common terminology in the discipline of enterprise architecture. In: Feuerlicht, G., Lamersdorf, W. (eds.) ICSOC 2008. LNCS, vol. 5472, pp. 400–413. Springer, Heidelberg (2009)
19. Wegmann, A.: The systemic enterprise architecture methodology (seam) - business and it alignment for competitveness. In: International Conference on Enterprise Information Systems 2003 (ICEIS 2003), pp. 483–490 (2003)
20. Op 't Land, M., Proper, E., Waage, M., Cloo, J., Steghuis, C.: Enterprise Architecture: Creating Value by Informed Governance, 1st edn. The Enterprise Engineering Series. Springer (December 2008)
21. Zachman, J.A.: A framework for information systems architecture. IBM Systems Journal 26(3), 276–292 (1987)
22. Schekkerman, J.: Trends in enterprise architecture: How are organizations progressing? Technical report, Institute For Enterprise Architecture Developments (2005)
23. Fatolahi, A., Shams, F.: An investigation into applying uml to the zachman framework. Information Systems Frontiers 8(2), 133–143 (2006)
24. Council, C.: A practical guide to federal enterprise architecture (2001)
25. Pereira, C.M., Sousa, P.: A method to define an enterprise architecture using the zachman framework. In: SAC 2004: Proceedings of the 2004 ACM Symposium on Applied Computing, pp. 1366–1371. ACM, New York (2004)
26. Buckl, S., Ernst, A.M., Matthes, F., Ramacher, R., Schweda, C.M.: Using enterprise architecture management patterns to complement togaf. In: IEEE International Enterprise Distributed Object Computing Conference, pp. 34–41. IEEE Computer Society, Los Alamitos (2009)
27. Mannaert, H., Verelst, J.: Normalized Systems—Re-creating Information Technology Based on Laws for Software Evolvability, Koppa, Kermt, Belgium (2009)
28. Mannaert, H., Verelst, J., Ven, K.: The transformation of requirements into software primitives: Studying evolvability based on systems theoretic stability. Science of Computer Programming 76(12), 1210–1222 (2011)

29. Mannaert, H., Verelst, J., Ven, K.: Towards evolvable software architectures based on systems theoretic stability. Software: Practice and Experience 42(1), 89–116 (2011)
30. Van Nuffel, D.: Towards Designing Modular and Evolvable Business Processes. PhD thesis, University of Antwerp (2011)
31. Huysmans, P.: On the Feasibility of Normalized Enterprises: Applying Normalized Systems Theory to the High-Level Design of Enterprises. PhD thesis, University of Antwerp (2011)
32. Baldwin, C.Y., Clark, K.B.: Design Rules. The Power of Modularity, vol. 1. MIT Press Books. The MIT Press (January 2000)
33. Richardson, G.L., Jackson, B.M., Dickson, G.W.: A principles-based enterprise architecture: lessons from texaco and star enterprise. MIS Quarterly 14(4), 385–403 (1990)
34. Lindstrom, A.: On the syntax and semantics of architectural principles. In: Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS 2006), vol. 8, p. 178b (2006)
35. Weinberg, G.M.: An Introduction to General Systems Thinking. Wiley-Interscience (1975)
36. Bertalanffy, L.V.: General Systems Theory: Foundations, Development, Applications. George Braziller, New York (1968)
37. Bunge, M.: Treatise on Basic Philosophy. Ontology II: A World of Systems, vol. 4. Reidel, Boston (1979)
38. Gero, J.S., Kannengiesser, U.: The situated function-behaviour-structure framework. Design Studies 25(4), 373–391 (2004)
39. Sutcliffe, A.G., Maiden, N.A.M.: Domain modeling for reuse. In: Frakes, W.B. (ed.) Third International Conference on Software Reuse, pp. 169–177. IEEE Computer Society Press, Los Alamitos (1994)
40. Smith, H.E.: Modularity in contracts: Boilerplate and information flow. American Law & Economics Association Annual Meetings Paper 46 (2006)
41. Yin, R.K.: Case Study Research: Design and Methods, 3rd edn. Sage Publications, Newbury Park (2003)

Research Approach in Enterprise Engineering: A Matter of Engineering

Niek J. Pluijmert[1,2], Wolfgang A. Molnar[3], and Hend*erik* A. Proper[3]

[1] Radboud University, Nijmegen
[2] INQA Quality Consultants B.V., P.O. Box 195, 4130 ED Vianen, The Netherlands
[3] CRP Henri Tudor, 29, avenue John F. Kennedy, L-1855 Luxembourg-Kirchberg
pluijmert@inqa.nl, email@wolfgang-molnar.com, erik.proper@tudor.lu

Abstract. Enterprises encounter serious problems in keeping pace with ever faster changing markets. Enterprise Engineering (EE) is an emerging field that is promising in providing solutions. Doing research in this field, requires choosing an appropriate research method for different parts of the research. This is the composition of the research method from known research methods, we call this engineering of the research approach. We structure available methods, approaches and techniques for qualitative research in information systems. We describe three epistemologies and discuss the different qualitative research methods and differences and similarities between them. For our research on EE that applies transaction cost economics in designing enterprises using the notions of Enterprise Ontology and Enterprise Architecture we combine a positivist approach during literature study with an interpretivist approach during Action Research.

Keywords: Enterprise Engineering, Research Approach, Qualitative Research Methods, DEMO, Design Science, Action Research.

1 Introduction

Organizations change rapidly because they must keep pace with ever faster changing markets. But organizations encounter serious problems in controlling changes. In trying to adapt to necessary changes, organizations may perform projects. In international surveys[1] it is found that many strategic projects fail. More in detail it is found that most of the large (labor costs over $ 10 million) projects fail or are challenged, while small (labor costs less than $ 1 million) succeed. The top 3 causes are (lack of) involvement of qualified users, support by management (fast decision making) and clear objectives. Tools and infrastructure are in the 10th place. So, technical factors (tools and infrastructure) are less important than social factors (the top 3 causes) for projects to be successful. Dietz & Hoogervorst [1] investigated to what extend enterprises derive success from their strategy: the majority of the strategic initiatives fail. They distinguish two factors, one is the strategy chosen and its adaptation over time and the other is the implementation

[1] International Standish Group 2010.

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 73–86, 2013.

of the chosen strategy. They argue that research has shown that strategic failure is mostly the avoidable result of inadequate strategy implementation. The impact of IT on enterprises is increasingly acknowledged to be fundamentally strategic as new IT penetrates progressively the core business processes [2]. So the integration of IT in the enterprise is necessary because then the emphasis is put on the social factors and less on technical factors.

An answer to these problems can probably be found with Enterprise Engineering (EE). EE is an emerging field and combines relevant parts of traditional organization sciences and information system sciences.

Chen & Vernadat [3] define EE as the art of analyzing, restructuring, designing - or redesigning - and, as much as possible, optimizing whole or part of a business entity with respect to its mission and objectives, where a business entity is any socio-economic system built to produce products or services. Vernadat [4] states that the current challenge is to build agile enterprises, which he describes as systems of cooperating business entities that can belong to different legal entities and that can be easily tailored to fast changing conditions. Business entities are combinations of business processes. Chen, Doumeingts & Vernadat [5] define Enterprise Architecture (EA) as the description of the basic arrangement and connectivity of parts of a system. They state that the concept of architecture is closely related to engineering.

According to Dietz [6] the basic premise of EE is that an enterprise is a designed system and for the design of an enterprise the notions of Enterprise Ontology (EO) and Enterprise Architecture (EA) are crucial. EO provides a means to make a model of the construction of an enterprises at a high level of abstraction, i.e. completely independent of its implementation. EO also has a precise definition of a business process, that consists of transactions. EA is defined as the set of design principles that an enterprise applies in designing itself. As is known from engineering practice, without these principles the design freedom would be practically unlimited which is of course not desirable.

Although the definitions of EE in [3,5] compared with the definition in [6] are different, their definitions of business process are very much alike. That is why we feel EE is a good candidate to offer solutions for the above mentioned problems enterprises encounter to keep pace with the ever faster changing markets.

Fields like EE require specific research methodologies with which one can devise artifacts and at the same time study how useful they are. The reason for this article is that we want to make clear (for the readers as well as for ourselves) the position that we take towards research in EE and what are accepted ways of doing research in this field. We base ourselves on the work of Myers [7] about qualitative research in IS which is available in an updated version as a website[2].

EE is about organizations and information systems (IS). We approach the research from the IS side and we realize we also could have taken an approach from the organization sciences. We take the approach from the IS side because it is about engineering and also because this fits best with our background, that is in IS.

[2] http://www.qual.auckland.ac.nz/

2 Research Objectives and Research Questions

The objective of our research is to better understand the role of transaction costs in designing enterprises. This is the operationalisation of our aim in Section 1 to make clear what accepted ways of doing research in the EE field are. The notions of EE, EO and EA as described in the previous Section 1 [3,4,5,6] provide the theoretical basis for designing enterprises as combinations of business processes. Costs are an important aspect in organizing an enterprise. Williamson [8] characterizes transaction by 3 dimensions and 3 governance structures. The dimensions are uncertainty, the frequency with which transactions recur and the degree to which investments are transaction specific. The governance structures are market, internal or an intermediate mode of organization. The characteristics of the dimensions determine where transactions can best be located: on the market, internal or in an intermediate mode. This theory can provide evidence to support the way transactions as defined in [6] should be combined into business entities that form the agile enterprises as defined in [3] that we need to face nowadays problems of ever faster changing markets. This leads to the central research question *how can agile enterprises be designed in practice by applying Transaction Cost Economics to Enterprise Engineering?*

3 Literature Review

First we will give the definition of Enterprise Engineering we will use and that comprises the definitions from [3,4,5,6]. In accordance with the AppEER- website[3] we define EE as the overarching term for the disciplines (among which are both EA and EO, see Section 1, as well as business process management, enterprise modeling, enterprise transformation) that study the engineering of socio-technical systems. With socio-technical systems we mean specifically information systems (IS) in their full alignment with their human / organizational context. Because EE is an emerging field of research without (yet) commonly accepted research approaches, we investigate in this chapter research approaches for IS research to find an approach for EE. There are different ways to structure methods, approaches and techniques that researchers use. We want to study organizational and managerial aspects of IS and we want to study in depth a limited number of cases. Because of these reasons quantitative (e.g. statistical) methods are not appropriate, so we will describe qualitative methods. In doing so, we follow Myers [7]. In Figure 1 we schematically give the relation between qualitative research methods and epistemology that we will explain. We start to discuss research approaches from the viewpoint of epistemology. Next, we take into account different methodologies and last we discuss differences between and possible combinations of the different methodologies.

[3] Appeer.ee-team.eu

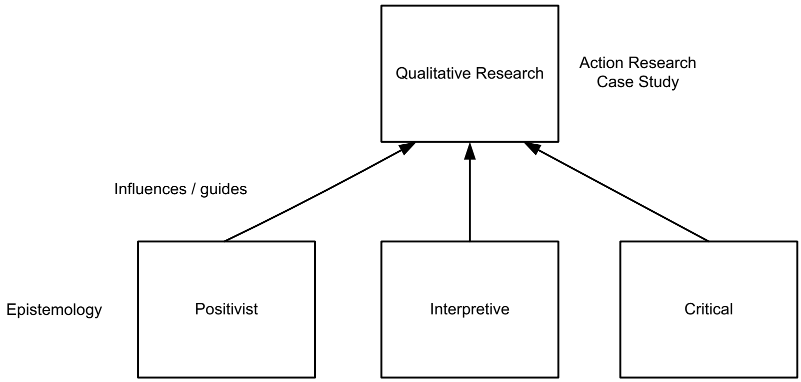


Fig. 1. Epistemology and Qualitative Research Methods

3.1 Epistemology

Epistemology is a branch of philosophy that studies the nature and scope of knowledge, more specifically it is concerned with what is acceptable knowledge in a certain field. Orlikowski & Baroudi [9] made an inventory of 155 information systems research articles published from 1983 to 1988. They examined the articles for the underlying epistemology and could classify the articles as positivist, interpretive or critical. This classification is also proposed by Chua [10].

Positivism. Positivist studies are primarily meant to test theory. Researcher and object of study are independent. Researchers assume an objective physical and social world that exists independent of human beings. This is the traditional approach of natural and social research [9]. Applied to IS research, it is applicable for situations where the designer wants to evaluate if the designed artifact works according to the specifications of the design. Because a positive approach does not take into account the effect on human beings, this approach is less suited for studying the effect of IS on the human beings that work with the IS.

Interpretivism. In interpretivism one asserts that reality is a social product and hence incapable of being understood independent of the social actors that construct that social reality. In interpretive research the researcher tries to understand how members of a social group by their participation in the social processes, help to constitute their social action. So, no objective reality exists and the perception and the importance of subjective meanings is emphasized. Applied to IS research, part of the evaluation is also the way the designed artifact works in its environment of human actors or, in other words, the judgment if the designed artifact works according to the user's requirements. The evaluation results can lead to improvements of the designed artifact.

Criticism. The critical researcher tries to critically evaluate and transform social reality, this is opposed to the other two research perspectives that confine themselves to predict or explain social reality. The idea within criticism is that social reality is historically constituted and that by critiquing existing social systems the contradictions and conflicts can be revealed so that people can act to change the existing system. Again, applied to IS research, the designed artifact is not only observed in its working environment, but the researcher also tries to influence the environment. At this point, we must discern between the following two things: first we have the content of the artifact and second we have the project that implements the artifact. The researcher can have two roles: the role of defining the content of the artifact and/or the role of project leader. The researcher can have the role of project leader in an interpretive or critical setting. Only if the researcher also determines the content of the artifact, we speak of a critical role for the researcher. Thinking of nowadays social reality, where IS is supposed to have a supportive role in organizations, one should be careful in applying the critical approach. This seems to be confirmed by Orlikowski & Baroudi [9] who classified none of the studied articles as critical.

3.2 Qualitative Research Methods

In the last part of Section 1 we wrote that we follow Myers [7] in his description of qualitative research in information systems. He argues that for studying social phenomena, a qualitative approach is most appropriate and in Section 2 we explained why this applies to our research. A research method is the strategy of inquiry that is used to design the research and collect data. This implies that the research method can be used with every of the three epistemologies. Here, we will discuss action research (AR) and case study research (CSR) as qualitative research methods. As we shall see, AR has much in common with design science (DS) research , we will also discuss DS research and the distinction between AR and DS research.

Action Rsearch. AR originally aims to contribute to the solution of immediate problematic situations of people and to the body of knowledge of the social science community. It had much difficulties in being acknowledged as a good research approach for IS. Baskerville & Wood-Harper [11] address the relationship between AR and consulting: action researchers are required to defend their method against the challenge that "this is nothing but consultancy!" At that time Baskerville recognized that AR was not a main stream social science technique that was transported to the IS field. He thinks that maybe in the IS field AR will finally flourish. In Baskerville & Myers [12] AR has proven its added value in IS research as a method to solve current practical problems while expanding scientific knowledge. The trigger for AR has been the frequent calls for IS researchers to make their research more relevant.

Case Study Research. The term case study has multiple meanings. It can be used to describe the unit of analysis or to describe a research method. As the unit of analysis, case study can be used in conjunction with e.g. AR. As a research method there are numerous definitions and Yin [13] defines the case study as a research strategy that attempts to examine (a) a contemporary phenomenon in its real-life context especially when (b) the boundaries between phenomenon and context are not clearly evident. Benbasat, Goldstein et al. [14] add to the definition of Yin that multiple methods of data collection are employed to gather information. They sum up a list of 11 key characteristics for case studies, see Table 1.

Table 1. Key characteristics of case studies [14]

1.	Phenomenon is examined in a natural setting
2.	Data are collected by multiple means
3.	One or few entities (person, group or organization) are examined
4.	The complexity of the unit is studied intensively
5.	Case studies are more suitable for the exploration, classification and hypothesis development stages of the knowledge building process; the investigator should have a receptive attitude towards exploration
6.	No experimental controls or manipulation are involved
7.	The investigator may not specify the set of independent and dependent variables in advance
8.	The results derived depend heavily on the integrative powers of the investigator
9.	Changes in site selection and data collection methods should take place as the investigator develops new hypotheses
10.	Case research is useful in the study of 'why' and 'how' questions because these deal with operational links to be traced over time rather than with frequency or incidence
11.	The focus is on contemporary events

Design Science. According to March & Smith [15] IT research studies artificial as opposed to natural phenomena, DS is concerned with devising artifacts to attain goals. They define a framework that consists of four types of products (constructs, models, methods and implementations) and four research activities (build, evaluate, theorize and justify). Build and evaluate are design science activities and theorize and justify are natural science activities. In a later article of Hevner [16] IS research is characterized by two paradigms: behavioral and design science. From the business-to-IT alignment model from Henderson & Venkatraman [17] the framework from Hevner is extended with a relevance part which provides requirements from the environment (business needs) for the IS research. In [16] IS research is defined as a three cycle method: relevance cycle and rigor cycle with the design cycle in the middle, see Figure 2. The goal of behavioral-science research is truth (relevance cycle). The goal of design-science research is utility (design cycle). In the rigor cycle the foundations and methodologies that together form the knowledge base for accomplishing IS, research are provided and new additions to the knowledge base are made.

Hevner [16] is a reaction to the essay of Iivari [18] in which is argued that the distinction between IS as DS and IS as inventions from practitioners, is the specification of a reasonably rigorous constructive research method for building IT artifacts.

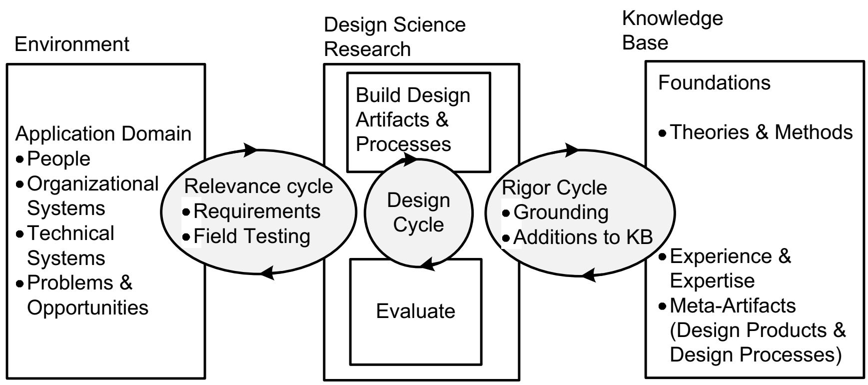


Fig. 2. Three Cycle view of Design Science, Hevner [16]

Peffers, Tuunanen et al. [19] developed a design science research methodology (DSRM), because they felt that the lack of such a methodology in IS research could have contributed to the slow adoption in IS. Peffers et al wanted to design a commonly accepted framework. Therefore they studied influential research and used a consensus-building approach to design the framework. This framework consists of six activities and has four possible entry points for research, see Figure 3.

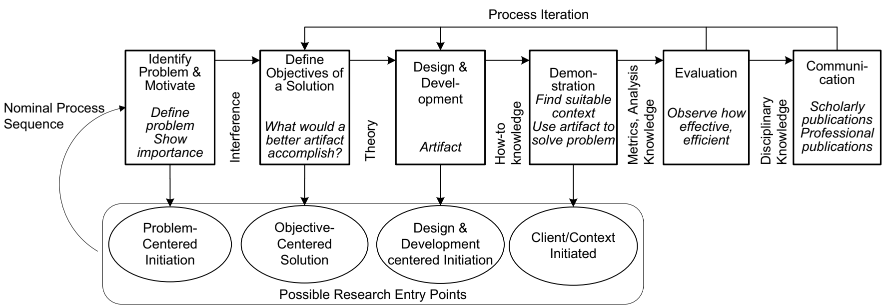


Fig. 3. Design Science Research Methodology, Peffers et al. [19]

3.3 Differences and Similarities between DS and AR

In literature discussion can be found whether AR and DS are similar or not. Iivari [18] differentiates between DS and AR: DS has its roots in engineering and AR has its roots in the socio-technical design movement. Iivari & Venable [20] argue that DS research and AR are decisively dissimilar: they discern on the one hand purely technical problems and innovations and on the other socio-technical problems and innovations. Iivari & Venable [20] argue that DS research focuses on constructing new and innovative ways to solve class(es) of problems, thus creating new reality. In many situations AR is conducted to understand phenomena like the working of complex organizational situations and human behavior and come to improvements to that. So, DS concentrates on constructing new artifacts, while AR concentrates on improving the use of artifacts in their meant environment. Järvinen [21] argues that AR is similar to DS. He does so by trying to collect the characteristics which describe the nature of AR in general on the one hand and the characteristics of DS on the other hand. He characterizes AR with 7 characteristics and DS with 6 characteristics. He concludes that it seems there is a very high fit between the two sets of characteristics. In table 2 the corresponding characteristics are placed opposite to each other. So we see that Järvinen and Iivari do not disagree so much, but that the involvement of the environment in the evaluation is seen as an important dissimilarity by Iivari, while Jrvinen considers this a similarity.

3.4 Combining AR and DS to Overcome Problems with Traditional AR

Iivari & Venable [20] describe also situations where AR and DS research can be combined. When we take in mind the nowadays practices of software development like Agile and Scrum and also a management method as Lean, it seems that a shift towards AR is driven by the environment's demand to respond faster [22]. They define IT artifacts as ensembles, by which is reflected that structures of the organizational domain are inscribed into the artifact during its development and use [22]. Orlikowski [23] identifies the prevailing DS approach as "build and then evaluate". AR aims at linking theory with practice, and thinking with doing. They propose to combine the two methods into a new method action design research (ADR) and in this way recognize that the artifact emerges from interaction with the organizational context. The method has 4 stages (problem formulation; building, intervention and evaluation (BIE); reflection and learning; formalization of learning) and 7 principles, see Figure 4. In the BIE-stage they discern IT-dominant BIE and organization-dominant BIE. In IT-dominant BIE artifacts are initially evaluated by practitioners and only more mature artifacts are evaluated by end-users. In organization-dominant BIE where the emphasis is on innovation in organizational intervention, each iteration is evaluated by both practitioners and end-users.

3.5 Project Management

The methods described before put emphasis on the *what* and pay less attention to the *how*. The *what* and the *how* correspond with the two roles of the researcher, *i.e.* the role of defining the content of the artifact and the role of project leader (see Subsection 3.1). A method that describes the *how* is that of Mulder in [24]. He applied AR and defined a project management method for AR projects, see Figure 5. He clarifies the coherence between a participative project approach and decision making in an organization. He found a large added value of this project management approach. Speed, quality and involvement of parties and by

Table 2. Similarities of the fundamental characteristics of action research and design science according to Järvinen [20]

Action Research	Design Science
AR-1: Action Research emphasizes the utility aspect of the future system from the people's point of view.	DS-4: Design science's products are assessed against criteria of value or utility.
AR-2: Action research produces knowledge to guide practice in modification.	DS-2: Design science produces design knowledge (concepts, constructs, models and methods).
AR-3: Action research means both action taking and evaluating.	DS-3: Building and evaluation are the two main activities of design science.
AR-4: Action research is carried out in collaboration between action researcher and the client system.	DS-5: Design science research is initiated by the researcher(s) interested in developing technological rules for a certain type of issue. Each individual case is primarily oriented at solving local problem in close collaboration with the local people.
AR-5: Action research modifies a given reality or develops a new system.	DS-1: Design science solves construction problems (producing new innovations) and improvement problems (improving the performance of existing entities).
AR-6: The researcher intervenes in the problem setting.	DS-5: Design science research is initiated by the researcher(s) interested in developing technological rules for a certain type of issue. Each individual case is primarily oriented at solving local problem in close collaboration with the local people.
AR-7: Knowledge is generated, used, tested and modified in the course of the action research project.	DS-6: Knowledge is generated, used and evaluated through the building action.

consequence the acceptance of the solution is by far better than a distant expert judgment. A participative approach considers organization design as a change process in which parties unfreeze from their current (undesired) situation, start discovering new possibilities for change and establish this in a new design.

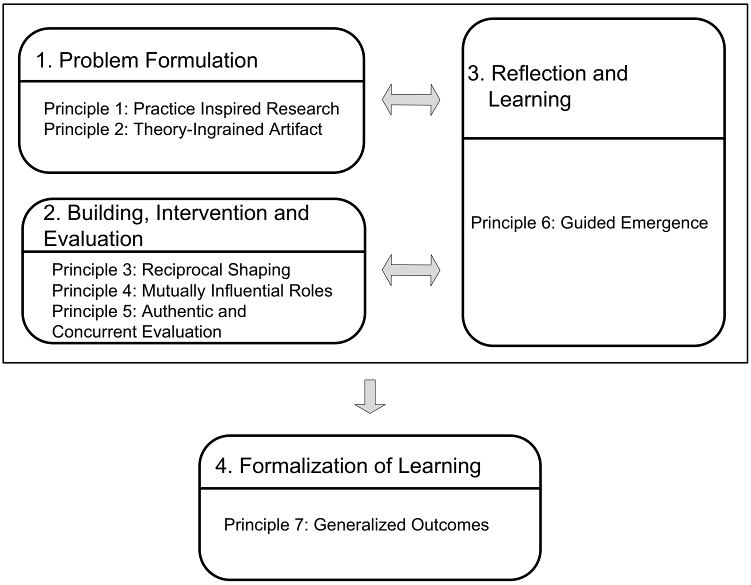


Fig. 4. Action Design Research (ADR) acc. Sein et al. [22]

4 Our Research Approach

In our research we will start studying theory and literature on transaction cost economics (TCE). The researcher takes in this a positivist position while performing the design cycle and the rigor cycle. This is a necessary step to take for us in order to ground our experiences from practice in theory and existing literature. This should lead to a framework for applying TCE to EE. We want to apply this framework in practice and learn from it. For this approach a qualitative research method is appropriate, not a quantitative. According to Orlikowski [9] and Figure 1 this means that we need an interpretive or critical stance. The critical stance implies that the researcher wants to change the social reality of the changing organization. This does not fit to a role as consultant as we have in our projects. In the project, we take decisions in the project management process and in the change process we propose the solutions and do interventions. According to Figure 1 this means that the researcher needs an interpretive role. As argued in Section 3 we follow [7,25] in using a qualitative approach in our research. We already discussed the differences and similarities between DS and AR [20,21,22,23] and we repeat here that an AR approach fits best in an Agile environment where the use of methods like Scrum and Lean is encountered. We use case studies for defining the unit of analysis. For the setup and execution of projects, we adopt Mulder [24], see Figure 5. In Table 3 we summarize our research approach.

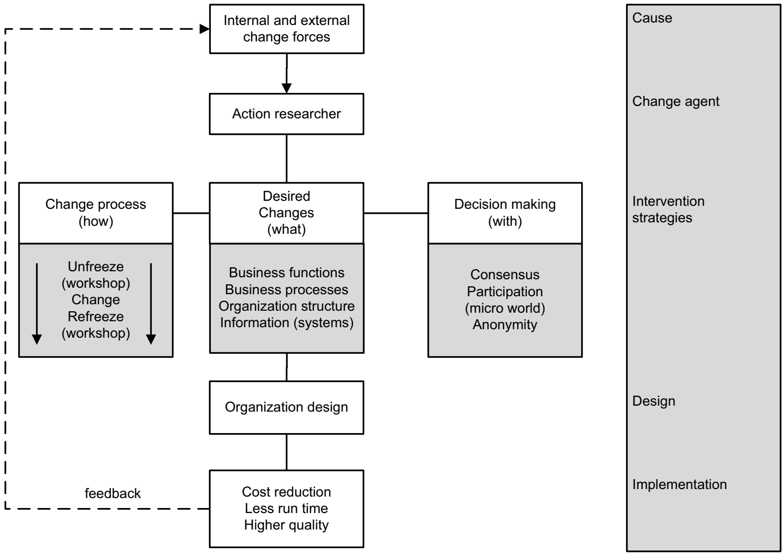


Fig. 5. Coherence between project approach and decision making in organization design [24]

Table 3. Summary of research approach

Objective	Enterprise design by applying transaction cost economics to EE
Focal level	Role of transaction costs in enterprise design
Epistemology	Positivist for theoretical study, Interpretivist in cases
Research Method	Action Research
Project Management Method	Participative

4.1 Activities Already Performed

In Section 1 we followed Dietz [6] that Enterprise Ontology (EO) is a crucial notion for EE. A method for EO is DEMO and that method has been applied in a number of projects. As part of this research, we investigated finished DEMO projects for what the added value of DEMO as an EE method in projects has been. A first round of interviews with the stakeholders of these DEMO projects and the founders of DEMO has been conducted in April – June 2012. The notion "DEMO project" means that the people involved in those projects acknowledge them as DEMO project. All projects investigated are claimed to be successful. After the first, more general, round of interviews, there will be a second round of interviews, that will concentrate on specific subjects that have been discovered during the first round as interesting for further investigation. The projects were diverse in scope, see Table 4. We developed an interview list with questions from different perspectives and with different questions for the different types of stakeholder. We will not discuss the results of the interviews here, but we confine ourselves to remarking that we are in the process of interpreting the interviews and defining the subjects for the second round of interviews.

Table 4. Projects investigated for their application of DEMO

Project name / project organization	Characteristics
VISI	Development of a model of large construction projects and a model for software for exchange of messages for coordination between project partners
KLM Air France	Choice of information system for the merger of the two cargo divisions
Rijkswaterstaat	Application Portfolio Rationalization
ING	Implementation of Shared Service Center Securities in a bank

5 Conclusions

We argued that it is important in research to discern between different aspects while doing IS research. The philosophical assumptions (epistemology) a researcher has, and the methods of doing research he uses, determine the scientific relevance of the things he does. In literature, researchers did try to make sharp distinction between different research methods and to prove that one method is better than another. Other authors tried to prove the similarities of different methods and also there are authors who try to combine different methods into one new method. We believe that it makes more sense to make a combination from different available approaches and justify why this combination is best suited for the situation at hand. Providing a solution for the problems enterprises encounter with ever faster changing markets, requires at first the study of theory and other literature with a positivist attitude. But in nowadays world of Scrum an Lean, applying theory right away in practice is required. For this, AR is an appropriate approach and the researcher needs an interpretivist stance in applying the solution provided.

References

1. Dietz, J.L.G., Hoogervorst, J.A.P.: Enterprise Ontology and Enterprise Architecture – how to let them evolve into effective complementary notions. GEAO Journal of Enterprise Architecture 1, 1–19 (2007)
2. Grgecic, D., Rosenkranz, C.: Information Systems Change and Social Interaction: A Research Agenda. In: 18th European Conference on Information Systems (2010)

3. Chen, D., Vernadat, F.: Standards on Enterprise Integration and Engineering–State of the Art. International Journal of Computer Integrated Manufacturing 17(3), 235–253 (2004)
4. Vernadat, F.: Reengineering the Organization with a Service Orientation. In: Hsu, C. (ed.) Service Enterprise Integration, pp. 77–101. Springer, US (2006)
5. Chen, D., Doumeingts, G., Vernadat, F.: Architectures for Enterprise Integration and Interoperability: Past, Present and Future. Computers in Industry 59(7), 647–659 (2008)
6. Dietz, J.L.G.: Architecture: Building Strategy into Design. Sdu Publishing (2008)
7. Myers, M.D.: Qualitative Research in Information Systems. MIS Quarterly 21(2), 241–243 (1997)
8. Williamson, O.E.: Transaction Cost Economics: The Governance of Contractual Relations. Journal of Law and Economics 22(2), 233–261 (1979)
9. Orlikowski, W.J., Baroudi, J.J.: Studying Information Technology in Organizations: Research Approaches and Assumptions. Information Systems Research 2(1), 1–28 (1991)
10. Chua, W.F.: Radical Developments in Accounting Thought. Accounting Review 61(4), 601–632 (1986)
11. Baskerville, R., Wood-Harper, T.: A Critical Perspective on Action Research as a Method for Information Systems Research. Journal of Information Technology 11, 235–246 (1996)
12. Baskerville, R., Myers, M.D.: Special Issue on Action Research in Information Systems: Making IS Research Relevant to Practice. MIS Quarterly 28(3), 329–335 (2004)
13. Yin, R.K.: The Case Study Crisis: Some Answers. Administrative Science Quarterly 65, 58–65 (1981)
14. Benbasat, I., Goldstein, D.K., Mead, M.: The Case Research Strategy in Studies of Information Systems. MIS Quarterly 11(3), 369–386 (2008)
15. March, S.T., Smith, G.F.: Design and Natural Science Research on Information Technology. Decision Support Systems 15(4), 251–266 (1995)
16. Hevner, A.R.: A Three Cycle View of Design Science Research. Scandinavian Journal of Information Systems 19(2), 87–92 (2007)
17. Henderson, J., Venkatraman, N.: Strategic Alignment: Leveraging Information Technology for Transforming Organizations. IBM Systems Journal 38(2&3), 472–484 (1999)
18. Iivari, J.: A Paradigmatic Analysis of Information Systems As a Design Science. Scandinavian Journal of Information Systems 19(2), 39–64 (2007)
19. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems 24(3), 45–77 (2007)
20. Iivari, J., Venable, J.: Action Research and Design Science Research – Seemingly Similar but Decisively Dissimilar. In: Newell, S., Whitley, E.A., Pouloudi, N., Wareham, J., Mathiassen, L. (eds.) 17th European Conference on Information Systems, pp. 1642–1653 (2009)
21. Järvinen, P.: Action Research is Similar to Design Science. Quality & Quantity 41(1), 37–54 (2007)

22. Sein, M.K., Henfridsson, O., Purao, S., Rossi, M., Lindgren, R.: Action Design Research. MIS Quarterly 35(1), 37–56 (2011)
23. Orlikowski, W.J.: Using Technology and Constituting Structures; A Practice Lens for Studying Technology in Organizations. Organization Science 11(4), 404–428 (2000)
24. Mulder, H.B.F.: Rapid Enterprise Design. PhD thesis, Technical University Delft (2006)
25. Klein, H.K., Myers, M.D.: A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems. MIS Quarterly 23(1), 67–94 (1999)

Development of Software Tool Support for Enterprise Architecture in Small and Medium-Sized Enterprises

Joost Dumeez, Maxime Bernaert, and Geert Poels

Department of Management Information Systems and Operations Management
Faculty of Economics and Business Administration,
Ghent University, Tweekerkenstraat 2, B-9000 Ghent, Belgium
{Joost.Dumeez,Maxime.Bernaert,Geert.Poels}@UGent.be

Abstract. Throughout recent years a lot of research has been done to develop enterprise architecture (EA) approaches for large and complex enterprises. Consequently, an array of tools has been developed for these large enterprises to aid in EA management. However, traditional small and medium-sized enterprises (SMEs), which are very important for economy, have to a great extent been neglected. Recently research has been done towards a new EA approach for SMEs. The approach is called CHOOSE. As tool support is almost indispensable in complex environments, the need for tool support was quickly experienced while doing case studies in SMEs. Unfortunately, tool support is already rated low on usability by EA practitioners in large companies. A different approach was required to provide tool support for managers in SMEs. The developed software tool already received positive feedback from managers.

Keywords: Enterprise architecture, software tool support, small and medium-sized enterprise, CHOOSE.

1 Introduction

A common analogy is often made between enterprise architecture (EA) and architecture in construction. When an architect is asked to design a house, the focus will be on where the staircase has to be or how many rooms or balconies there have to be. The specifics of the color or brand of paint are not of interest yet. The future habitant agrees on some kind of blueprint, a master plan that will serve as the starting point for more detailed decisions. This high level blueprint will show the major functions of the house and how these have to be constructed. As future occupant of the house this high level and abstract representation is probably the most informative. On the other hand, the engineer or constructor is probably more interested in a detailed elaboration.

When companies became bigger and more complex the need for EA arose. The advantages of EA for large enterprises have been widely acknowledged in practice and literature [1]. Landenberg and Wegmann [2] investigated which aspects of EA were being researched. One of their conclusions was that interest in the field is currently growing, given the increase in publications. Furthermore, Ernst et al. [3] found in their survey that in practice, EA is also of growing importance. However, most of this research is targeted towards architecture in large enterprises. But while small and medium-sized enterprises (SMEs) are considered the backbone of our

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 87–98, 2013.

economy [4], little research has been done towards EA approaches for those SMEs. In a literature research concerning information systems in SMEs [5], not one paper was found from 1979 to 2008 about EA in SMEs.

In response to this lack of research about EA for SMEs, Bernaert and Poels proposed a new EA approach for SMEs [6; 7]. A metamodel was created, focusing on the essential dimensions and characteristics of EA in the context of SMEs. The metamodel was called the CHOOSE metamodel, which is an acronym for "keep Control, by means of a Holistic Overview, based on Objectives and kept Simple, of your Enterprise".

Just as tool support is widely considered indispensable in large companies, the need for it was also quickly identified in SMEs by several case studies. However, as existing tools are already considered low on usability by EA practitioners, a different approach is proposed in this research: a software application for tablets. As SMEs usually rely on insight from the CEO and do not have any enterprise architects on staff, this software application had to be useable for management.

In the next chapter, the problem statement and goal of this research are discussed. In chapter three, existing tool support is first considered. In the fourth chapter, our solution towards a user-friendly tool for the EA approach CHOOSE is then proposed. The last chapter concludes with a summary and future research directions.

2 Problem Statement and Goal of the Research

2.1 Specific Research Problem

While developing the EA approach CHOOSE, case studies were performed in SMEs. Even after the first tests, it was clear that tool support was indispensable. Working on a whiteboard with post-its was not practicable (Fig. 1).

Fig. 1. Need for tool support during case studies

First of all, input was very slow, having to write every element on a separate post-it. If a goal had to be corrected, a new post-it had to be made. Once the model was finished, it was impractical to store away three whiteboards and present it on a meeting the week after. More importantly, finding an element could take a while and making any sort of analysis was out of the question. Academic sources also widely support the necessity of software tool support.

"For EAs to be useful and provide business value, their development, maintenance, and implementation should be managed effectively and supported by tools." [8]

"EA management should be supported by tools, which support distributed access to consistent data, offer the possibility to structure the information managed, and also aid users in filling out their role in the EA management process." [3]

3 Related Research

3.1 Existing Tool Support

Kurpjuweit and Winter [9] identify three different objectives for which tools can be used. They can be used to document and communicate, they can be used to analyze and explain or they can be used to design the EA. Typically after selecting an EA approach and selecting the preferred modeling language, there are software tools available to model your enterprise. Several options are available on the market, for example IBM's system architect [10] or Metis from Troux technologies [11]. However most of these tools are focused on the design aspect and are solely used to model the enterprise.

Based on a survey performed by the university of Munich, Ernst et al. summarize how different tools perform according to different criteria and different scenarios [3]. They came up with a kiviat diagram, showing the minimum, maximum, and median score obtained by the different tools (Fig. 2). We can see that even for EA practitioners, the average usability of the different tools scores rather low. One reason was that all tools came with predefined metamodels. Some came with up to 400 entities with corresponding associations. Even for daily practitioners this is perceived to be very complex, making the tools hard to work with in practice. Especially the visualization needed improvement. Two major issues occur when visualizing an EA with existing tools. First, the automated generation of adequate visualization is mostly not possible. The second issue is that the semantics of those visualizations are often not properly defined. This leaves a lot of room for interpretation. These issues are both explained by the simple fact that most tools are drawing tools where visualizations are manually created. This often causes practitioners to use certain symbols or links out of context. Although the model is understandable for the creators of the model, the tool can no longer interpret the results correctly. The result is a drawing rather than a model. A nice quote from this paper that summarizes this paragraph: "drawing is no management".

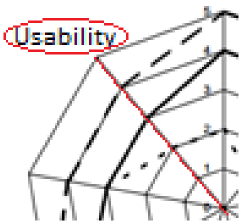


Fig. 2. Diagram showing a low usability rating

The graphical user interface of a typical EA tool is usually composed of three different components (Fig. 3 from left to right). The content explorer, a canvas to do the modeling itself, and a concept explorer with the different modeling language constructs. Often this works by dragging and dropping the different constructs from the concept explorer to the canvas and thus slowly building up the different models.

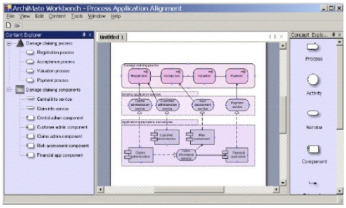


Fig. 3. Typical EA tool

4 Solution Approach

While it has been made clear that tool support is needed, one of the key words in the abbreviation CHOOSE was ‘Simple’. Given the resource poverty of SMEs [12], they are unable to staff an enterprise architect, or pay expensive consulting fees. As described above, modeling tools are typically aimed towards enterprise architects. So while developing a tool for this new approach, it was clear that contrary to the above tools, a different approach would be required for SMEs. Even practitioners in big companies rated the existing tools rather poor in usability. Furthermore these tools expect the users to be fully aware of the models and modeling languages.

To find an approach towards tool development, a methodology that is aimed to achieve user-friendly interfaces was adopted from Cooper et al [13]. He was one of the first to criticize traditional software development processes and was the first to pioneer a software development process that was based on the users’ needs. How could a good product be created when it did not take into account the users’ goals, needs and motivations? The improved software development process that he proposed now emphasizes the design aspect (Fig. 4).

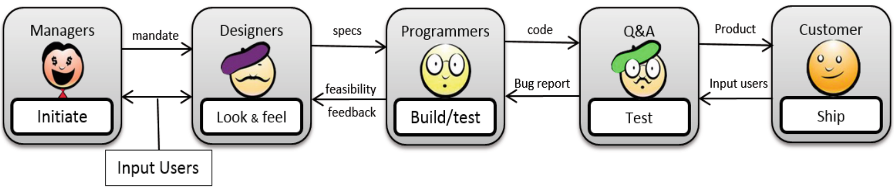


Fig. 4. Software development process

The first step, called ‘initiate’, is the step where the need for a certain program is identified. The need for tool support has already been discussed as being necessary to support an EA approach. The necessity of this new EA approach has also been shortly dealt with in the introduction. As the tool is currently still under development (build phase), the last two steps in the process will not be dealt with in this paper. The technicalities of the building itself are also beyond the scope of this paper. In the next section we will thus focus on the design step. An approach towards design is introduced in the next section.

4.1 Goal-Directed Design

The task of the design step in Fig. 4 should not only be about the appearance of a product. When properly deployed it should identify user requirements and take into account his/her behavior. Design should be about product definition, based on the goals of users, needs of the business and the constraints of technology. Design is defined in a broader sense.

Just as we made an analogy with architecture when arguing that EA is needed, a similar reasoning can be made here. In the case of an architect, he/she will have to understand how the people live and work before he/she can start designing a house. He/she then should sketch the spaces to support and facilitate those behaviors. For the architect, designing a house is more than making the house look pretty.

This broader approach towards design is called 'interaction design'. Interaction design is more than making sure something looks pretty; interaction design is about understanding the users and knowing cognitive principles.

An approach to implement this design philosophy is proposed by Cooper et al. [13]. In short, this approach is called 'goal-directed design'. The main steps followed in a goal-directed design approach are shown in Fig. 5.

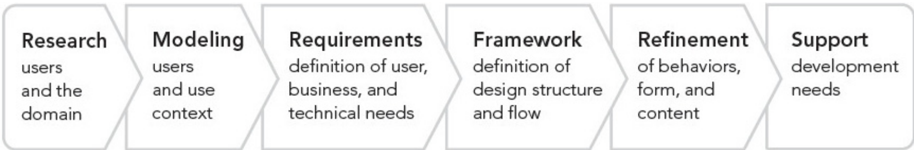


Fig. 5. Goal-directed design process

Research. The research phase consists mainly of understanding the future user and knowing who that future user will be. In the research done to develop the CHOOSE approach it was already clear that the CEO would have to be closely involved. There are two reasons illustrating the need for the CEO's involvement.

Firstly in SMEs, employees generally do not know the structure, let alone know why they are doing something. It is generally only the entrepreneur who knows the whole working of the company and this distinguishes SMEs from larger companies, because a CEO can have an overview of its SME. Secondly, in SMEs, the job description for employees is often vaguely defined and an overview of tasks or responsibilities is often missing [14].

In a study performed by Hankinson et al. [15], managers responded that they worked between 50 and 60 hours a week. Most time was spent in meetings and informal interactions (50%) and on the phone (40%). The remainder was spent travelling. Only 5 hours or less than 10% of time was used on strategic issues, personal analysis, and analyzing results. Notably, managers generally indicated that their time management was poor. To get a better insight in the lifestyle of the manager, a small amount of managers was asked to keep a very detailed schedule of their workweek. In this small-scale test, the results, stated above, could be verified. We clearly saw that managers were very busy with day-to-day activities. This left little time in between for working on strategic issues [16].

Another important factor to keep in mind is that managers are not expert technology users. According to [17], users can be divided in three types: experts, willing adopters and mainstreamers. Mainstreamers are people who do not use technology to use the technology. They use it to get something done. They use just a few key features and do not care about any other.

Modeling. In the modeling step, a persona is defined, which is the portrait of what the ideal user would be. Additionally a list of goals of what the user is interested in is composed. This is usually input from the research step. By completely defining a 'persona', programmers and designers always keep focus on who they are making the program for. Usually, programmers talk about 'the end user'. By making the end user an actual person, it restricts them of stretching the 'end user' to fit the situation. 'This is easy' might be true for the programmers, but may not be the case for the defined persona, and obliges programmers to rethink their approach to fit the persona's needs.

Because a complete persona definition is beyond the scope of this paper, a small summary is offered. Our target user is a manager in an SME, who is generally not technology savvy. He/she often does not know about the specifics of EA. Though, if he/she would use an EA approach, he/she would want it to be simple while still offering value. It would have to aid in keeping an oversight of the company.

Requirements Definition. While developing the CHOOSE approach, a list of eleven main criteria were found for EA in an SME context [7]. As the tool is used to support this approach it should at least be able to achieve these criteria. These criteria will serve as a guideline when setting requirements for the tool.

Requirements for tools can be thought of as consisting of objects, actions, and contexts. For example, for a calling application this might be something like: 'Call (action) a person (objects) directly from an appointment (context)'. As it can be quite difficult to extract information in this format, it is easier to separate them into data, functional and contextual requirements [18].

Data Requirements. These requirements are the objects and information that our users want from our tool. As can be identified by the criteria from EA, we need data output that can give an overview of the company. Furthermore a description of how things are done in the company is needed. A focused view will provide a more detailed view. This output furthermore has to be usable in other often-used programs like Excel and PowerPoint.

Functional Requirements. A mobile application was chosen to fit the busy schedule of a manager. Since there is little time to work on strategic matters during the day, the only adoption of an application would happen if it could be used during downtime. It has to be useable when they have a few minutes spare time in a waiting room or maybe five minutes before they go to bed. Showcased by the popularity of the iPad and other Android tablets, tablets are on the rise. Even in SMEs, it is becoming the standard tool that every manager has with him/her [19]. A tablet was chosen over a mobile phone because it offers almost the same mobility while giving more screen estate.

Contextual Requirements

- Business Requirements: It has to be cheap to implement [12].
- Experience Requirements: If we want managers to spend their precious time on EA and use this tool, it has to be appealing and fun.
- Technical Requirements*:* It has to work on a platform that can complement a manager his/her usage pattern. As the CHOOSE metamodel is still being refined, the software architecture should be flexible to allow changes to be adapted quickly.

Summary of Tool Criteria. The following list of five criteria is proposed for this tool. These will be used as guidance when developing the application:

1. Offer a focus view of enterprise components
2. Offer a holistic overview of the firm
3. Simplicity, the tool has to be easy to use
4. Fit in day-to-day activities of manager
5. Fun and appealing

Design Framework. Considering usability was an issue with existing tools for EA practitioners, making the tool usable for managers would be challenging. If we go back to the roots of usability, Vitruvius is considered as being the first student of ergonomics and usability [17]. His questions were based around three key words:

- Firmitas: How durable is the design and does it have strength?
- Utilitas: How useful is the design and does it fit the user's needs?
- Venustas: How beautiful is the design?

What Vitruvius did not suggest was how to achieve these objectives. However, directions to achieve these objectives are offered by Davidson et al. [20]. They found an important link between mental models and usability.

Norman [21], an academic researcher in interface design, argues that design is often constrained by culture. People want to be able to foresee what an object will do. When an interaction with a system is deeply engrained through cultural learning it is hard to change the way people interact with an object. The easiest way to make an interface design consistent with how a user expects it to behave is by mimicking the physical product. For example, calculator applications or calendar applications often mimic the behavior of the physical products. Unfortunately, as is the case with our application, there is not always a physical object that can be mimicked. The insight to decrease the discrepancy between how an object behaves and how a user expects it to behave is offered by Norman [22], Cooper et al. [13] and IBM [23]. They have discerned three models of a system:

- Mental Model: How a user perceives that a product works.
- Implementation Model: Actual way the system works internally (programmer's perspective).
- Represented Model: The way the application is represented to the user.

According to Davidson et al. [20], who researched this link between usability and mental models, the program will be perceived easier to use and easier to understand the closer the Represented Model resembles the Mental Model.

Several design methods are used to support and influence these mental models of the user: simplicity, familiarity, availability, flexibility, feedback, safety, and affordance are proposed by IBM [24]. By following these methods, the user's mental model will be better aligned with the represented model. According to [20], the application will then have higher usability.

Simplicity has been one of the key words throughout the development of the CHOOSE approach (the S stands for Simple). Additionally it came out as one of the main criteria during the requirements definition. As simplicity is the main guideline, in the next section this is the only design method that is further zoomed into. A design framework is furthermore also concerned with detailed design to make the application look beautiful. Although beauty is subjective, a lot of guidelines were taken into account, though are not further elaborated here and will be presented in a thesis.

Simplicity: Less Is More. When visiting a website it can sometimes feel like walking through Times Square: You see a myriad of different advertisement billboards each with catchy slogans like "The next big thing", "Probably the best beer in the world", "The real thing". Many of these billboards are very flashy with vibrant colors and flashing lights, trying to get the passer-by its attention. Every brand competes with each other to make the billboard that pops out the most, trying to capture the so important passer-by glance.

However, people are impatient and they can only process a limited amount of information [24]. The more content you show, the smaller your chances will be of them to notice your most important content. Less really is more in this case.

Even small details like words, colors, buttons can add to the already heavy load a person has to process. Removing irrelevant options, slogans, and content decreases this load on users. Designing a user interface in a clever way can further diminish this load and enhance user experience [17].

Thompson et al. [25] did an experiment to find out whether users preferred features over usability or vice versa. They concluded that features sell a product better, but when users were able to test the product before their purchase they would buy the product with better usability over the product with more features. In software, the cost of adding extra features is close to nothing. Considering people appear to buy products based on features, this leads to features piled upon features. The problem is that with each extra feature that is added, the product becomes more difficult to use.

Iyengar and Lepper [26] set up multiple experiments to test whether people like more or less choice. All results obtained showed significant higher sales revenues when offering only a handful of options. They also found that people where more satisfied with what they bought when being offered fewer options. Because it offers a person a sense of control, people prefer to have a choice over no choice at all. However if that choice exceeds a handful of options, choice can get overwhelming.

4.2 Tool Presentation

During the requirements definition stage in the goal-directed design process, five criteria were derived for our tool (Section 4.1.3). Furthermore simplicity, familiarity, availability, flexibility, feedback, safety, and affordance were the design methods that were introduced in the design framework (Section 4.1.4). While presenting the tool, some criteria and design methods will be referred to where they are applicable.

One aspect of the CHOOSE approach is to build up a goal tree. To showcase the tool, an extract from a case study is presented. The case study was performed in an SME that sells tires and does some maintenance on cars. One of the goals of this company was to increase its customer base. Two possible scenarios were considered. The first scenario was opening a new shop, the second was to increase visibility through improved signing of the building and better online visibility. To showcase the tool, the goal 'online visibility' is added to the goal tree (Fig. 6).

When starting up the application, the user is presented with a playful start screen. At first glance the start screen is already radically different from traditional approaches. Firstly it has to entice the manager of working with the program when he/she has some spare time (Criterion 5). Another important criterion is that no outside help is needed to exercise EA (SMEs' resource constraint). At the bottom right a question mark is shown. Whenever in doubt about some concept, the manager can quickly get an explanation (Design method: safety). In our case we head over to the goals part of the application and start the wizard for adding a new goal. In step 1 the name of the goal and a description can be added. In the following screen, since 'online visibility' is one of the goals to achieve a general increase in visibility, the latter goal is selected as the upper goal. Next we want this goal to be assigned to the marketing expert. This is done by browsing through the actor tree and selecting our marketing expert. Next we need a process to operationalize this goal. Here 'manage social media' is selected to operationalize 'online visibility'. Finally 'Facebook' is selected to be the online platform of choice. Facebook will serve as input and output for our operation 'manage social media' and is thus also of concern for our goal 'online visibility'. Finally, in the last step, a small summary is shown before the goal can be saved. When we return to the 'goals' part of the application, we can now browse to the goal 'online visibility' (Fig. 7). In this view, the file explorer from Microsoft Windows is mimicked which should make it very intuitive for users to browse through the goals tree (Design method: familiarity). At the right hand side a focused view of all the connected elements is offered (Criterion 1). Note that very few buttons are offered, providing just enough features (Criterion 3). The main button here is the add button at the top right corner. This started the wizard that guided us through the procedure of adding our goal 'online visibility'. By guiding the user step by step, information is shown in chewable chunks, so it never becomes overwhelming for the user (Criterion 3). Every model can be searched through in the same way, offering a consistent user experience (Criterion 3).

(Prototype: https://www.dropbox.com/sh/03vyn790i1wggsv/G_uDPwJy00/CHOOSE)

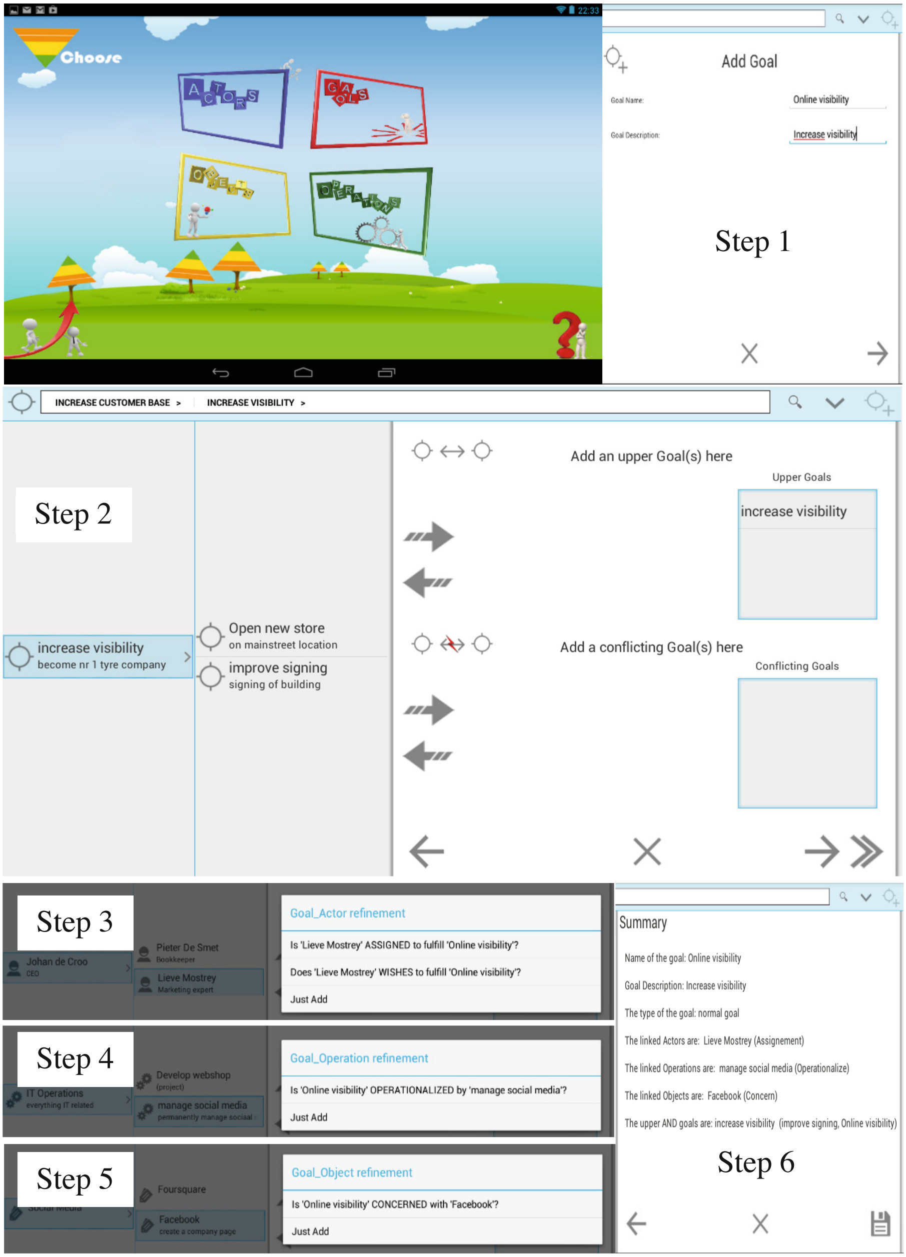


Fig. 6. Add goal 'online visibility'

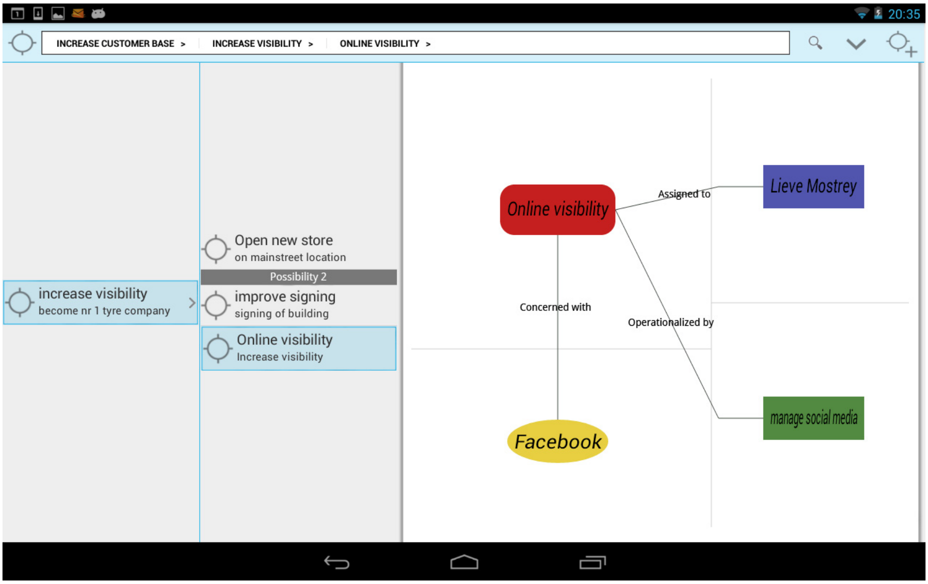


Fig. 7. Focused view 'online visibility'

5 Conclusion

EA approaches have been primarily developed for large companies, despite the importance of SMEs for economy. Tools furthermore have been developed to aid enterprise architects. In this paper, we investigated how tool support could be made usable for managers in SMEs. By using a goal-directed design approach in the tool development process, we came up with a radically different application. A tablet application has been proposed that is easy enough for managers and CEOs to use. First tests in SMEs have been very promising. They found it much easier to comprehend then traditional tools and were very enthusiastic about using a tablet. Further evaluation in practice will still be needed to evaluate the tool and the EA approach CHOOSE.

Although we focused on CEOs and managers working in SMEs, the goal-directed design approach could be expanded towards larger companies. Further research could find ways of better involving other stakeholders with easy to use applications. For example, employees could use a smartphone application to get an overview of their tasks and responsibilities, connected to the companies' goals. This could offer employees a sense of purpose.

References

1. Whittle, R., Myrick, C.: Enterprise Business Architecture: The Formal Link between Strategy and Results. IBM Systems Journal 1, 32–51 (2004)
2. Langenberg, K., Wegmann, A.: Enterprise architecture: What aspects is current research targeting. Laboratory of Systemic Modeling (2004)

3. Ernst, A.M., et al.: Tool support for enterprise architecture management: strengths and weaknesses. In: 10th IEEE International EDOC, pp. 13–22. IEEE Press (2006)
4. European Commission: Are EU SMEs recovering from the crisis. Annual Report on EU Small and Medium sized Enterprises (2010)
5. Devos, J.: ICT-bestuur voor kleine en middelgrote ondernemingen (2011)
6. Bernaert, M., Poels, G.: The quest for know-how, know-why, know-what and know-who: using KAOS for enterprise modelling. In: Salinesi, C., Pastor, O. (eds.) CAiSE Workshops 2011. LNBIP, vol. 83, pp. 29–40. Springer, Heidelberg (2011)
7. Bernaert, M., et al.: Enterprise Architecture for Small and Medium-sized Enterprises: a Starting Point for Bringing EA to SMEs, Based on Adoption Models. In: Devos, J., Van Landeghem, H., et al. (eds.) Information Systems and Small and Medium-Sized Enterprises (SMEs): State of Art of IS Research in SMEs. Springer, Berlin (2013)
8. Schekkerman, J.: Enterprise Architecture Tool Selection Guide v5.0. Institute for Enterprise Architecture Developments (2009)
9. Kurpjuweit, S., Winter, R.: Viewpoint-based meta model engineering. In: EMISA Workshop, pp. 143–161 (2007)
10. IBM System architect, `http://www-01.ibm.com/software/awdtools/systemarchitect/`
11. Troux Metis, `http://www.troux.com/products/troux_software/`
12. Beck, T., Demirguc-Kunt, A.: Small and medium-size enterprises: Access to finance as a growth constraint. J. Bank Financ. 30(11), 2931–2943 (2006)
13. Cooper, A., et al.: About face 3: the essentials of interaction design. Wiley (2012)
14. Kamsties, E., et al.: Requirements engineering in small and medium enterprises. Requir. Eng. 3(2), 84–90 (1998)
15. Hankinson, A., et al.: The key factors in the small profiles of small-medium enterprise owner-managers that influence business performance: An international research project UK survey. Int. J. Entrep. Behav. Res. 3(3), 168–175 (1997)
16. Malhotra, R., Temponi, C.: Critical decisions for ERP integration: Small business issues. Int. J. Inform. Manage. 30(1), 28–37 (2010)
17. Colborne, G.: Simple and Usable Web, Mobile, and Interaction Design. New Riders (2010)
18. Shneiderman, B., Ben, S.: Designing the user interface. Pearson Education, India (1998)
19. Rising popularity of tablet use in small business community, `http://www.forbes.com/sites/thesba/2012/04/03/rising-popularity-of-tablet-use-in-small-business-community/`
20. Davidson, M.J., et al.: Mental models and usability. Cognitive Psychol. 15 (1999)
21. Norman, D.A.: Affordance and design, `http://www.liacs.nl/~fverbeek/courses/hci/AffordancesandDesign.pdf`
22. Norman, D.A.: The design of everyday things. Basic Books (2002)
23. IBM: Object-Oriented Interface: IBM Common User Access Guidelines. Design (1992)
24. Hertwig, R., Todd, P.M.: More is not always better: The benefits of cognitive limits (2003)
25. Thompson, D.V., et al.: Feature fatigue: When product capabilities become too much of a good thing. J. Marketing Res., 431–442 (2005)
26. Iyengar, S.S., Lepper, M.R.: Rethinking the value of choice: a cultural perspective on intrinsic motivation. J. Pers. Soc. Psychol. 76(3), 349 (1999)

Towards the Definition, Implementation and Communication of an IT Strategy: The Case of IT Strategy at EPFL

Gorica Tapandjieva[1], Didier Rey Marchetti[1], Irina Rychkova[2], and Alain Wegmann[1]

[1] École Polytechnique Fédérale de Lausanne,
Systemic Modeling Laboratory LAMS
Station 14, CH-1015 Lausanne, Switzerland
{gorica.tapandjieva,didier.rey,alain.wegmann}@epfl.ch
http://www.epfl.ch

[2] Centre de Recherches en Informatique, Paris 1
90 rue Tolbiac, C.14.05, 75013 Paris, France
irina.rychkova@univ-paris1.fr
http://www.univ-paris1.fr/

Abstract. Large enterprises need to coordinate the IT initiatives that exist in different organisational units of the enterprise. If these initiatives are not coordinated, the resulting IT system is likely to become difficult to use and expensive to develop/maintain. Enterprise architecture methods are designed for that purpose. We report on the use of a service-oriented enterprise architecture method, called SEAM, in the context of a mid-size university. The originality of SEAM is its service orientation and the recursive modeling from business down to systems. Using SEAM, we develop a service model of the overall organisation. The model is stored in a web-based tool. We also propose a concrete implementation of architectural principles described in the literature. This principles help build an integrated IT system. The paper explains the background of this project and the current progresses. This approach illustrates how enterprises can build a common view for their IT resources.

Keywords: enterprise architecture, services, business and IT alignment, SEAM, Solu-QIQ.

1 Introduction

Today organisations rely on information technology (IT) as a support for their business goals, regardless of the services and products they provide. Business and IT alignment play a vital role in an organisation's success. IT provides business value, so the IT strategy should be carefully developed together with the business strategy.

All the work presented here is based on a case study. The case study is the enterprise architecture (EA) of the École Polytechnique Fédérale de Lausanne

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 99–110, 2013.

(EPFL). On the main campus there are over 11,000 people including students, researchers and staff. The work of all these people is supported by IT systems. Some of these IT systems are designed for all of the people on campus, some are used only by students and professors, and some are used only by the administrative staff.

In general, IT strategy includes multiples aspects spanning from technology selection to business process structure. In this project we are focusing mostly on how to best structure the relations between the business processes and the IT services, through the implementation of the architecture principles proposed by Ross, Weill and Robertson [6].

Our goal is to illustrate our proposed IT strategy. We present the goals and the plan of actions through working on this case study at EPFL. First, in Section 2, we focus on understanding the challenges. Then, in Section 3, we list the initiatives at EPFL that currently address these challenges. In Section 4, we present the basic elements that construct the IT strategy based on EA principles [6], and we suggest a way of communicating the IT strategy within the organisation. For this communication we use the Solu-QIQ tool, based on the methodology of urbanisation of IT systems (URBA), and the systemic enterprise architecture methodology (SEAM). In Section 5, we illustrate the application of our suggestions on a specific example of the business process, the process of hiring new Ph.D. students. We finish with the overall conclusions and future work.

2 Main Challenges

EPFL is a large organisation (with over 4500 employees) and it faces many IT challenges. For example, currently at EPFL there are around 100 different systems (both business and infrastructure) that provide services to EPFL's community. They can be grouped in several categories. Some of these systems are listed in Table 1.

The table shows systems managed by the Central Services. In addition to these systems, each laboratory within an institute or a school, has its own resources. The data about IT systems at EPFL is gathered by EPFL's information system's coordinator, Didier Rey. From this data, we see that all of the resources involved in the operation, maintenance and development of IT systems are highly distributed.

Additionally, new emergent technologies regularly appear. When used, they can cause changes in the current landscape of systems. In terms of resources, the situation becomes more complicated. The number of new IT projects continuously increases, as well as the time required to complete them. A major challenge is to manage even the new projects with the current resources. Some of these resources are the most basic ones: electricity and cooling.

Furthermore, EPFL is a complex system that has an impact by delivering services that support its three core missions: education, research and technology transfer. To ensure the quality of service (QoS), EPFL must protect its boundaries and be efficient inside this boundary. We can achieve this efficiency by optimizing the business processes and the resources used.

Table 1. Some of the IT systems at EPFL, grouped in several categories

	Mission:
IS-Academia	Academic management tool used by students, lecturers, researchers, administrative staff, etc.
Infoscience	Database used to archive and report works and scientific publications.
Moodle	Learning management system and course sharing platform used as course support.
GrantsBD	Portal for uploading and following research and scientific equipments requests at EPFL
	Administration:
SAP	Finances and human resources management system
IS-Academia	The same academic management tool can be used for room booking, courses and exams schedule, grades management, etc.
Sharepoint	Content and document management system
	Infrastructure:
Active Directory	Directory service.
VPN	Virtual private network.
CAMIPRO	Systems around EPFL student (or staff) card with electronic chip, that allow people to use a variety of functionalities like: payment in the miscellaneous campus shops and stores without any ready cash, access to some protected rooms, borrowing a bike, etc.

Indeed, the current IT systems work well. The greatest challenge appears when business processes need to be integrated across organisational units. We see integration as a problem, because people and IT systems involved in business processes are typically not in one organisational unit. When people and IT systems are in separate units, it is difficult for them to work together and communicate.

EPFL has already made efforts to address all these challenges. These efforts are explained in the following section.

3 Current Initiatives

At EPFL exists an information systems coordination body (Coordination des systmes d'information – **CSIN**) that has the mission of aligning the development of IT systems with EPFL's strategy and of assuring the coordination between different IT units. This body has several committees, commissions and management groups. **CoordAppl** – Coordination des applications is one of them.

CSIN organised several workshops at EPFL with target to:

- define high-level strategy by reviewing the impact of the applications on the return of investments (Gartner, early 2012)
- define the set of principles on how to select a project by giving specific axioms (2011)

– provide project management training (Hermes training, 2011)
– build a prototype to evaluate the integration of IT systems, by using the open source enterprise service bus (WSO2 ESB) [11], a part of a master's thesis project.
– propose a service-oriented approach to a help desk (ITIL transition workshop at DIT, in progress)
– identify gaps in the existing processes and find solutions for them (CoordAppl SEAM workshop on Ph.D Hiring, 2012 – present)

The conclusions from these workshops apply only to their specific problems and do not offer long-term solutions for the overall IT landscape. We propose an approach that offers a durable solution for the organisation of IT systems at EPFL. The concrete example upon which we elaborate our work is based on the progress and outcome of the ongoing CoordAppl SEAM workshop. In this workshop, we work with people responsible for the main IT systems and the heads of certain IT departments at EPFL. The combination of their knowledge, expertise and experience helps us to identify problems and solutions.

In most of the workshops, we use SEAM [10,9,7], a family of methods for strategic thinking, business / IT alignment, and requirements engineering. We use SEAM to communicate about the organisation and the IT systems. Furthermore, SEAM conceptualizes different aspects of an organisation: its business, enterprise architecture and software development. Also, it shows different organisational-level hierarchies, and it enables designers to choose how to see the system: as a whole (a black box), or as a composite (a white box). In addition, SEAM shows the behaviours of the systems, the properties representing the information exchanged, and the actors participating in the process. The originality of SEAM is in the integration of generic system thinking principles into discipline-specific methods. [8]

4 Basic Elements of the Proposed IT Strategy

In the aforementioned workshops, we address the gaps inside business processes themselves and the gaps between business processes and IT. We followed Ross's, Weill's and Robertson's [6] recommendations of using **architectural principles**. These architectural principles "provide a rallying point for managers responsible for building out and exploiting the enterprise architecture"[6], which consists of the following:

1. *Core business processes*
2. *Shared data driving core processes*
3. *Key linking and automation technologies*
4. *Key customers*

The IT strategy we implement is our interpretation of these architectural principles. We explain:

1. *Core business processes* – For each core business process, there has to be someone who takes care of the user's needs, who knows every detail of the process, its rules and regulations. This person should be able to design the process and know how to change it when the business strategy changes. This is the role of a **process owner** [4].
2. *Shared data driving core processes* – Data is an essential asset of every organisation. Its integrity, quality, security and utility are the most important features. The person responsible for the data used and exchanged in a business process is the **data steward** [3].
3. *Key linking and automation technologies* – Our key concept is **service**. We base our understanding of a service on a definition from systems thinking, [2] "The service concept represents a unit of essential functionality that a system exposes to its environment"; where a system, "is a group of interacting, interrelated, and interdependent components that form a complex and unified whole". Based on the given definition, the key linking and automation technologies represent the connection of IT systems (done by a middleware) and the connection of services supporting processes. From this, the people (and systems) who participate in providing one service or process should belong to one (virtual) unit.
4. *Key customers* – As many people are involved in the business process, it is very important to identify the key **actors**.

In Section 2, we have seen the challenges EPFL faces. In this section, we propose a solution that requires changes in EPFL's enterprise architecture, which involves identifying a process owner, a data steward, key customers and a linking technology around the services. The change would not take place all at once, especially not when a large number of people and systems are involved. Educating people about this proposed strategy is an important part of a successful change. By learning about it, people can better accept the change. Once they see that their individual goals are aligned with the business goals and they observe their contribution to the strategy, the process of educating them should be an easy task. Therefore, applying the architectural principles in EPFL's IT strategy requires two actions:

1. **Create a service-oriented enterprise architecture (EA) based on the architectural principles presented.**
   At EPFL, there are over 50 separate ongoing projects that are part of the current IT strategy. Having synergy among these projects would improve the overall IT strategy. We use our interpretation of the architectural principles to achieve this synergy. As an enterprise is a complex system, its main characteristic is continuous evolution [9]. We manage this evolution by using SEAM representation of the architectural principles. In the various SEAM models, an enterprise's evolution and the changes we propose are easily reflected. We use SEAM because it is systematic and it is recursive from business down to technology. In a traceable manner SEAM explicitly shows the services provided and the processes that implement these services. Moreover, SEAM

models service systems allowing to modify the organizational boundaries. All SEAM models are made with the SeamCAD tool.

2. **Create a shared vision of the service architecture by communicating about this vision with everyone involved.**
   We use, together with SEAM, another EA approach: the urbanisation of IT systems (URBA). URBA is a methodology used mainly in France. We apply URBA's suggestion only for using a *cartography tool*. For this, we chose **Solu-QIQ** [1], a tool successfully used by many companies (including RATP [5]). The tool offers an iterative approach for building a cartography of the information systems. As an output, the tool automatically generates a navigational web site that we use to communicate with people involved in building the systems. The advantages of Solu-QIQ over other tools is that it can be used for massive modelling and that the meta-model in the background makes it fully customisable. This customisation allows Solu-QIQ to support SEAM, so we use it as a database to store our SEAM models.

5 Illustrating the IT Strategy: Example

IT strategy is a very broad field. When many systems are involved, as is the case with EPFL, the IT strategy has several dimensions. This is the reason we focus only on specific parts and domains of EPFL's business. The IT strategy we suggest can be applied, however, to all domains and functionalities. We show the work we did on one business process; it can serve as an example for working on other processes.

In this section, the spotlight is on the process of **hiring a new doctoral (Ph.D.) student**. As mentioned before, this process is used in the CoordAppl workshop. We use a SEAM model to better perceive the anomalies and problems in the process.

This process is useful for evaluation because is a common process at EPFL. From the Ph.D. hiring process, we learned which services and sub-processes can be also used in post-doc and professor hiring processes. Moreover, the problems (encountered by the actors in the process) demonstrate the need for integration of systems, alignment of IT with business process, and alignment of processes with business strategy.

In Subsection 5.1, we describe the flow of specific actions in the process. In Subsection 5.2, we identify the specific problems. This is followed by the SEAM representation of the as-is and the to-be models, shown in Subsection 5.3. In Subsection 5.4, we illustrate the basic functionalities of the Solu-QIQ cartography tool. In addition, we present an example of the service view. This view is important for communicating with people about our service-oriented EA vision of the IT strategy. We complete the section with suggestions on how to apply the IT strategy based on the lessons learned from working on this example.

5.1 Description of the Ph.D. Hiring Process

Inscription:
The process begins when *an applicant* fills an application record on IS-Academia (Table 1). Record validation is possible only if three reference letters are completed on IS-Academia directly by the *applicant's references.* Afterwards the *doctoral program committee* (including the *doctoral program assistant*) analyses all application records and decides who is admissible to the program. The doctoral program assistant informs the applicants about this decision by e-mail. The doctoral program assistant also informs the *lab professors* by e-mail.
Selection:
After having identified potential *(admitted)* applicants, the professor organises interviews (with or without the help of the *lab's administrative assistant*). After the interviews, matches are made with one professor and one student interested in working together. This part of the process is not supported by any existing system. IS-Academia is used only to insert some notes about candidates the professors are interested in, and of course, for reading the candidate's data from the application record.
Employment:
The professor informs the lab's administrative assistant of his decision, and this assistant asks the doctoral program assistant to prepare an *admission letter* that confirms the hiring of the selected candidate by the professor's lab. This letter has to be signed by the *doctoral program director* and the professor hiring the candidate. This is the key step in the actual hiring of the Ph.D. student. The lab administrative assistant, asks the future students for the usual required documents (CV, passport copy, etc.). These documents, together with the admission letter are sent to the *HR assistant*, who is responsible for making the contract and arranging for visa, if needed. Once the contract is ready, it is sent for signature to the future Ph.D. student and a new record in SAP is made for him. From this point on, the employment record of the Ph.D. student is in SAP, and his academic record is in IS-Academia.

5.2 Ph.D. Hiring Process: Problems

We identify, together with the CoordAppl workshop participants, most of the problems. These problems have two facets. The first facet shows the difficulties actors are facing. These actors are those directly involved in the process (professor, applicant, i.e., future Ph.D. student, doctoral school administrative assistant, HR assistant, etc.). Some of the problems are:

- Physical documents get lost during the process.
- The same data from the future Ph.D. student's record is inserted by different people in two separate systems: IS-Academia and SAP.
- Confidential data is easily accessible.
- After the applicant has been selected, he is not aware of the status of the process, so he cannot know when something goes wrong.

- Getting the hard copy of the contract is critical for a future Ph.D. student. He needs it to apply for housing and permit, if needed. Also, he must be officially enrolled in the doctoral program to benefit from free language courses.

The second facet shows the business and technical reasons these problems are difficult to solve:

- Different administrative procedures and rules in EPFL's 18 doctoral programs.
- No document management system connected with IS-Academia and SAP.
- Low level of coordination between the people involved in the separate units.
- Not integration between IS-Academia and SAP, leading to redundancy and duplication of data and documents.
- No person responsible for the business process, someone who knows the rules in depth and is able to make decisions (business process owner). The same applies to the data (data steward).

5.3 Systemic Enterprise Architecture Methodology (SEAM)

We use the SeamCAD tool to build the SEAM models that conceptualizes the business process. Fig. 1 shows the as-is model of the Ph.D hiring process. We do not show the details about the execution of the process and the documents exchanged. What can be seen is the separate organisational units that provide services and participate in the process. The problems identified are caused by lack of integration between these units.

The solution to this example problem is depicted on Fig. 2. Everyone providing services to the process is put in one virtual integrated unit. The SEAM representation of being in one container (called working object) means that everyone shares resources and is aware of what is going on inside. A + sign on an object represents what is added by the solution, and a $\sim$ sign represents what needs to be changed. Now, in the to-be model we have the following:

- a virtual Ph.D. hiring service unit is added, which groups everything involved in providing services to the Ph.D. hiring process;
- an additional role of a business process owner is attributed to the EDOC deputy dean;
- a data steward is added;
- an enterprise service bus (ESB) is added that links processes and services from different systems in the virtual Ph.D. hiring service unit;
- all the heads of the current organisational units are now added to the management of the integrated process.

Fig. 2 shows the service organisation of the Ph.D. hiring process. The only remaining challenge is moving people from the as-is to the to-be, hence we use the cartography tool.

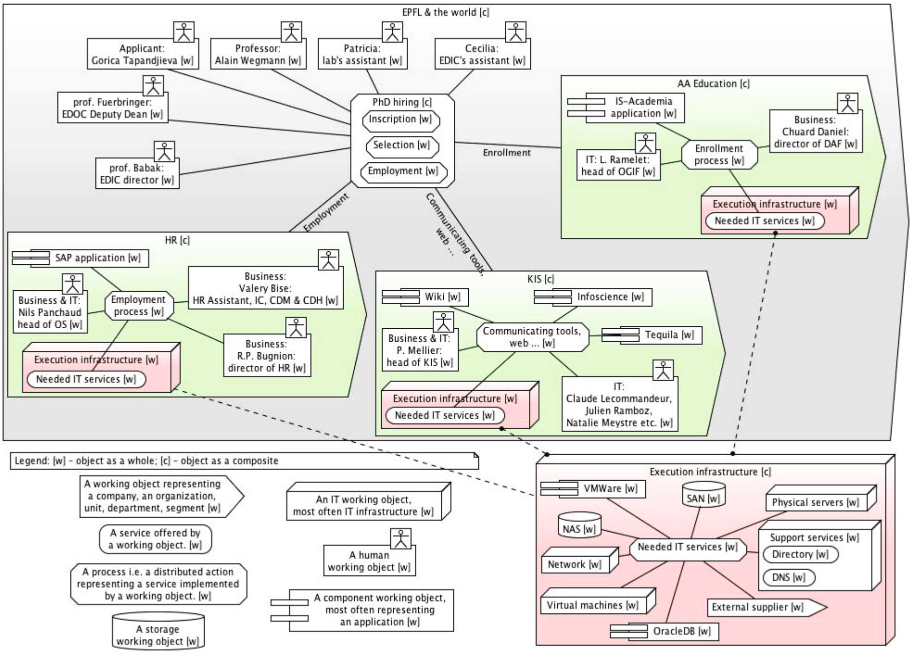


Fig. 1. SEAM as-is model of the Ph.D. hiring process. The separate organisational units can be seen.

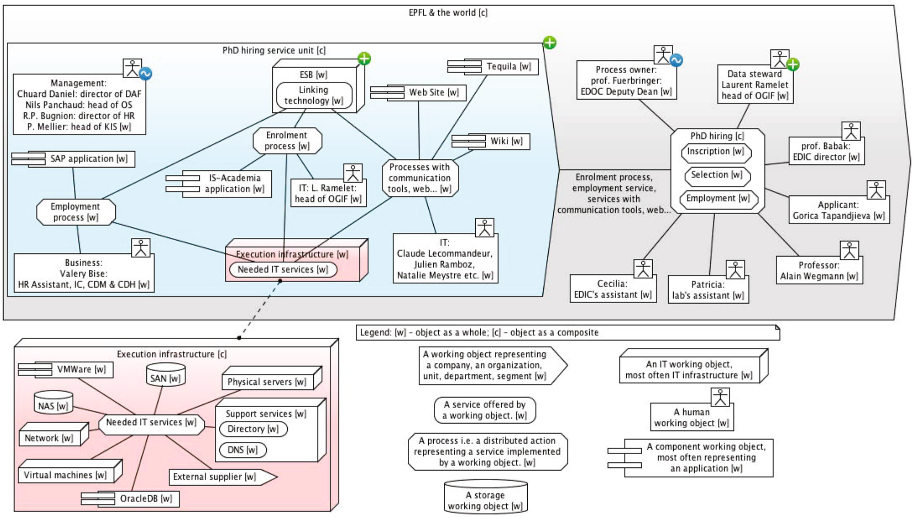


Fig. 2. SEAM to-be model of the Ph.D. hiring process. The virtual service unit integrating people and systems from several organisational units can be seen.

5.4 Solu-QIQ - Urbanisation of IT Systems (URBA)

The Solu-QIQ [1] cartography tool outputs a navigation web site which makes our IT strategy easily accessible and transparent. The first step is to define a meta model, the graphical representation of organisation's structure, showing the four basic urbanisation views: business, functional, application and infrastructure. In our meta model we additionally have a service view and an organisational view.

The meta model is only a database schema, so data has to be inserted into the tool in order to get an output. Data gathering requires interviewing involved people (business-process actors, department heads, IT specialists, etc.).

In order to ensure that the views of the participants have been correctly captured, additional in-depth interviews are necessary. Subsequently, the output is regenerated and rechecked. An example output[1] of the tool can be seen in Fig. 3. It shows a concrete example of a SEAM model: the IT systems (*IS-Academia and Moodle*) and actors (*Alain Wegmann and Didier Rey*) that provide a service (*Recuperation des fichiers*), projects related to this service (*GED and Support évolution...*), and the process in which the service is involved (*Sélection des candidat-e-s*).

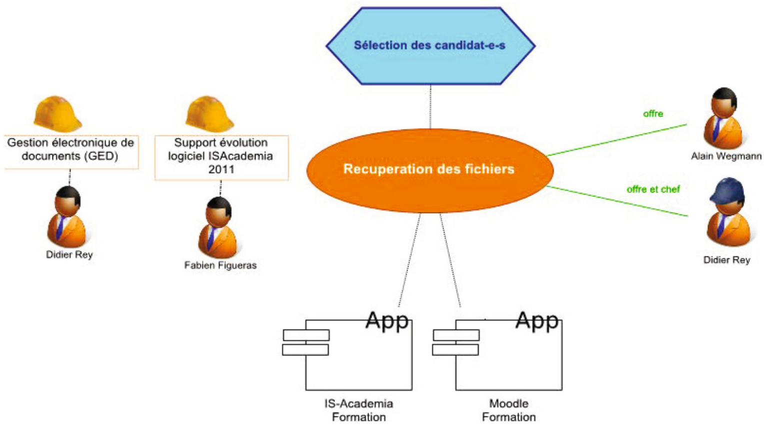


Fig. 3. One output of the web site that the Solu-QIQ tool generates, showing the applications and actors providing a service, projects around this service, and the process that consumes that service

5.5 Suggestions

From the Ph.D. hiring example, we are able to provide guidelines and describe the future work for employing the IT strategy. Here are our suggestions:

[1] All of the work connected with the cartography and the Solu-QIQ tool is done as a master's thesis project by a LAMS student.

1. Construct the SEAM as-is model following a pattern based on the example shown here. Communicate with all people involved in providing a service by using Solu-QIQ's output. Through interviews, all of these people contribute to making the cartography of the IT systems. This way, they see all the people and all the IT systems involved in providing the same service. If the output does not show their view of the service, they have the power to react (during the interviews) and even modify the shape of the cartography. After several iterations of interviews, the cartography output will converge to one common view. This is the most accurate view of the EA. Then, we can construct the SEAM to-be model, identify the process owner and the data steward. For the linking technology, once an appropriate middleware is found, it can be used for other business processes too. With these actions we suggest **what** to do.
2. Reorganise the organisational units into one virtual unit based on the services they provide (one unit per service). We recommend service-orientation in the whole enterprise, captured by SEAM usage. After the cartography is completed, actors can continue communicating by using SEAM models. This set of actions explain **why** we do what we do.
3. Obtain relevant information about certain IT systems (their use and interaction) by working on one functionality at a time. We work only on the processes that belong to that functionality. After the models have converged, we choose another business process belonging to that same functionality. This action explains **how** we plan to work across whole EPFL.
4. This work is currently done by a research lab (LAMS) acting as consultant, but later it will be performed by someone within the governance organization (CSIN). Appropriate person needs yet to be identified, so only in the future we can specify **who** will do the work.

6 Conclusions

In this paper, we first present the main challenges that EPFL faces with designing an IT strategy and the initiatives currently taken to overcome them. Then we address EPFL's enterprise architecture, according to our proposed IT strategy and our interpretation of the architectural principles. These architectural principles require that a process owner, a data steward, key linking technology and key actors are identified. We suggest service-oriented enterprise architecture and organising the units around the services they provide. These units should use a linking technology for integrating the different systems interacting in the process.

After defining this IT strategy, we need to communicate and share the vision. This is achieved by using SEAM and the tools SeamCAD and Solu-QIQ.

Using the Ph.D. hiring business process as an example, we describe the application of our proposed IT strategy. In several figures we show the SEAM as-is and to-be models. We illustrate a sample output of the Solu-QIQ tool and suggest steps to be taken for working on other processes.

From the overall work done so far, we conclude that the communication and the adoption of our approach is the most crucial part of the project.

7 Future Work

We have presented how to optimize one business process within EPFL. But an overall IT strategy will be complete only if we address all core business processes. For the future work, we propose to iteratively follow the suggestions from Subsection 5.5 for the remaining processes.

In parallel, to ensure the QoS of the provided services, we must also focus on protecting EPFL from its environment. This mainly includes working on several security standards and protocols.

Finally, until now, we haven't done a validation of our approach, but we have the full support of EPFL's CIO to carry on with this project as a mean to have visibility and transparency on the services defined, the services' providers and the services' users. The validation will happen after the Solu-QIQ tool is used routinely and after we gather the first feedbacks from the people involved.

References

1. AB+Software, `http://www.abplussoftware.fr/`
2. Anderson, V., Johnson, L.: Systems thinking basics. Pegasus Communications (1997)
3. Firth, C.: Data quality in practice: experience from the frontline. In: Conference of Information Quality, pp. 25–26. John Wiley and Sons Inc. (1996)
4. Hammer, M., Stanton, S.: How process enterprises really work. Harvard Business Review 77, 108–120 (1999)
5. Régie Autonome des Transports Parisiens, `http://www.ratp.fr/`
6. Ross, J.W., Weill, P., Robertson, D.: Enterprise Architecture As Strategy: Creating a Foundation for Business Execution. Harvard Business Review Press (2006)
7. Rychkova, I., Regev, G., Le, L., Wegmann, A.: From Business To IT with SEAM: the J2EE Pet Store Example. In: 11th IEEE International Enterprise Distributed Object Computing Conference, EDOC 2007, pp. 495–495. IEEE (2007)
8. SEAM Introduction, `http://lams.epfl.ch/reference/seam/`
9. Wegmann, A.: On the Systemic Enterprise Architecture Methodology (SEAM). In: Published at the International Conference on Enterprise Information Systems 2003, ICEIS 2003, Citeseer (2003)
10. Wegmann, A., Kotsalainen, A., Matthey, L., Regev, G., Giannattasio, A.: Augmenting the Zachman Enterprise Architecture Framework With a Systemic Conceptualization. In: 12th International IEEE Enterprise Distributed Object Computing Conference, EDOC 2008, pp. 3–13. IEEE (2008)
11. WSO2 Enterprise Service Bus, `http://wso2.com/products/enterprise-service-bus/`

Simulation-Driven Approach for Business Rules Discovery

Biljana Bajić-Bizumić[1], Irina Rychkova[2], and Alain Wegmann[1]

[1] École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
[2] Centre de Recherche en Informatique Université Paris 1 Panthéon - Sorbonne, 90, rue Tolbiac, 75013 Paris, France
{biljana.bajic,alain.wegmann}@epfl.ch, irina.rychkova@univ-paris1.fr

Abstract. Business rules are everywhere. Some of these rules are implicit and thus poorly enforced, others are written but not enforced, and still others are perhaps poorly written and obscurely enforced [1]. In this work, we propose an interactive, simulation-driven approach for the discovery of business rules. The rules are first specified in a natural language, then translated to the Alloy specification language. The Alloy Analyzer tool is used as a platform for rule simulation and discovery: it provides a domain specialist with an instant feedback, helping her to detect the issues with the existing business rules and to discover new rules in a systematic way.

Keywords: Business rules, Business Rule Discovery, Alloy, Requirements Elicitation.

1 Introduction

This paper illustrates how business rules for the order processing activity at Générale Ressorts SA can be discovered using the Alloy Analyzer tool [2]. The order processing example is academic, but it is grounded on a real problem: many insurance companies have a strategy for leveraging on late payments to maximize their return. In other industries, it can take up to three years between the ordering of parts and the payment. These terms are not captured as business rules, but they could be. When the IT systems make these payment terms explicit, management has to face many annoying facts in their business strategy. Another example: very often processes are defined in a strict way to make sure that the interest of the company is protected. But this is feasible only for second and third tier customers. Strategic customers always bypass the rules. When a rule is defined explicitly - the exceptions are not considered until a problem occurs with a key customer.

We claim that systematic capturing of business rules and the analysis of issues created by rules will help companies to define and improve their strategies.

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 111–123, 2013.

However, systematic handling of the rules expressed in a natural language is challenging: "Capturing the logic of an entire business would require probably many thousands of rules; a smaller subsystem, perhaps several hundreds" [3]; they can be specified by different analysts, can be inherited from the previous process versions, and can reflect different policies or strategic decisions.

The goal of this paper is to illustrate how Alloy can assist in the systematic discovery of business rules and the issues related to them. Our approach is based on rule modeling and simulation in Alloy Analyzer [2]. The use of formal specifications and model checking techniques allows us not only to *discover* the new rules but also to validate their consistency.

The remainder of this paper is organized as follows: In Section 2, we discuss our motivation and present the related works; In Section 3, we present our example - the Order Processing, specified for Générale Ressorts SA. We also introduce the Alloy specification language and discuss how business rules can be specified with this language. In Section 4, we present the Alloy model for Order Processing. In Section 5, we illustrate the BR discovery with Alloy Analyzer . We generalize our approach in a form of four steps to interactive BR discovery. In Section 6, we present our conclusions.

2 Motivation and Related Work

According to [3], "A business rule is a compact statement about an aspect of a business. It's a constraint, in the sense that a business rule lays down what must and must not be the case. At any particular point, it should be possible to determine that the condition implied by the constraint is true in a logical sense; if not, a remedial action is needed."

Many research and industrial publications are focused on challenges associated with business rules (BRs). In industry, vendors such as ILOG (currently a part of IBM), FICO Blaze Advisor and Pega Systems, Inc. have been developing business rule engines (BRE) since the late 1980s and are now leaders in the emerging BRE segment [4,5,6]. In academia, the computer sciences and engineering outlets have been active in business rule research [7,8,9], with extensive studies in rule programming, meta-modeling, rule mining, rules engines, business user interfaces and their role in services oriented architectures (SOA). Furthermore, joint academic and industry developed Object Management Groups (OMG) Semantics of Business Vocabulary and Business Rules (SBVR) standards (released in September 2006), which is intended to provide standards surrounding BR structure, terminology, classifications and meaning in BR authoring and repositories [10].

Different approaches propose different phases in business rules management life cycle (BRMLC). In [3] the main phases are discovery, definition, review and maintenance. By [1] the main phases are: discovery, analysis, design, authoring, validation, deployment. In [11] these phases are: plan, capture, organize, author, distribute, test and apply. In this paper, we focus on discovery (i.e. capturing) phase of BRMLC. The goal of discovery phase is to identify the potential business rules affecting the domain segment in development.

To see the effect of the introduced business rules, the designer must go through a number of phases, such as analysis, design, authorizing, validation and deployment. It would be much more effective, if the designer could get instant, visual feedback on how new business rules influence the behavior of a process. Examples of the company behavior could help him find what constraints are missing in the model. To discover new BR in an interactive way and ensure the consistency and validity of the overall set of BR, we propose the approach based on Alloy simulation.

Although BRs traditionally are expressed in a natural language [12], the works presented in [13] and in [10,14] report on other forms of BR formalization. In [13], the diagramatic language is used and in [10,14,15] the rules are specified with formulas in modal logic. In this work, we use an Alloy specification language (based on first order logic) and propose a technique for BR discovery based on model simulation and analysis in the Alloy Analyzer tool.

3 Modeling Business Rules with Alloy

In this section, we introduce our working example, the order processing, specified for Générale Ressorts SA. First, we specify the order processing and its associated business rules in a natural language; then, we discuss how this example can be specified in Alloy [16]. In the following sections we present the Alloy model for order processing and illustrate how the business rules for order processing can be interactively analyzed and discovered using the Alloy Analyzer tool [2].

3.1 Case Study: Order Processing in Générale Ressorts

Générale Ressorts SA is the market leader in watch barrel springs and a first-class manufacturer of tension springs, coil springs, shaped springs and industry components [17]. Générale Ressorts SA works with thousands of customers and strives to ensure the highest quality both for its products and for its customer services.

Order processing is one of the strategic activities in Générale Ressorts SA: it covers a complete order life cycle, from order creation to payment and delivery. Whereas the company constantly improves its technological processes in order to shorten the production cycle, the payment can take months after the product is delivered[1]. Therefore, flexible business rules for order processing and customer transactions management are essential for GR.

Order processing includes the following processes: order creation, order preparation, shipping and accounting. It is also closely related to the customer management processes in the company. The whole process, from the moment the customer makes an order to the delivery and the accounting is known as the order-to-cash cycle.

[1] The "shipping after payment confirmation" policy is not acceptable for this industry in general and for Générale Ressorts SA in particular.

In this paper, we define the (simplified) order processing activity that focuses on order creation, delivery and payment only: A customer submits an order request for manufacturing a watch component (part); the confirmed order is then prepared and delivered to customer. As stated above, the payment for the confirmed customer orders is a necessary condition to finalize the overall order processing transaction for a given order, though it is not required for order delivery.

Below, we present list of business rules related to order processing:

1. *Order creation*
   BR1.1 A customer order can be created and confirmed only for the customers registered in the enterprise information system.
   BR1.2 A customer order can be created and confirmed only for the parts existing in the product catalog.
   BR1.3 If an order request from a new customer is received, this customer has to be registered in the enterprise system.
2. *Order delivery*
   BR2.1 Every confirmed customer order must be eventually delivered to the customer.
3. *Accounting*
   BR3.1 Every confirmed customer order must be eventually paid by the customer.
4. *Customer management*
   BR4.1 Every customer record must contain one customer name.
   BR4.4 Every customer record must be associated with a previous orders history.
   BR4.5 A customer whose transactions with GR is equal or superior to XX XXX euro per year receives a status of strategic customer at GR.
   BR4.6 A customer whose transactions with GR is inferior to XX XXX euro per year receives a status of regular customer at GR.
   BR4.7 Strategic customers must always be able to submit the order with GR.

3.2 Alloy

Alloy [18] is a declarative specification language developed by the Software Design Group at MIT. Alloy is a language for expressing complex structural constraints and behaviour based on first-order logic.

The Alloy Analyzer [2] is a tool for the automated analysis of models written in the Alloy specification language. Given a logical formula and a data structure that defines the value domain for this formula, the Alloy Analyzer decides whether this formula is satisfiable. Mechanically, the Alloy Analyzer attempts to find a model instance - a binding of the variables to values - that makes the formula true. [19]

The syntax of Alloy is similar to the syntax of OCL (the Object Constraint Language) for UML [20]. In the following lines, the Alloy keywords are marked in bold. Data structures are represented with signatures (**sig**) and fields. Logic of Alloy language combines the quantifiers of first-order logic ($\exists$ (**one**), $\forall$ (**all**), etc.) with the arithmetic operators ($+, -.$ $=$,etc.), set operators ($\cup$, $\cap$, $\subset$ (**in**), etc.), relational and logical operators ($\neg$ (**not**/!), $\wedge$ (**and**/&&), $\vee$ (**or**/||), $\Rightarrow$ (**implies**/=>), etc.).

There are three types of constraints specified in Alloy: Fact (**fact**) is a model constraint that permanently holds; Predicate (**pred**) is a constraint that holds in specific context or for a specific part of the model only; Assertion (**assert**) is a property that the designer believes should be implied from the model and can check (command **check**) if it can be deduced from the other (permanent or contextual) constraints. In the examples given in the paper, assertion is presented in a shorter form **assertionName: check**. In our previous work [21] we defined the iterative process for service design where the Alloy signatures, facts and predicates were used for service specification. In this work, we extend the use of Alloy constructs: in particular, we use Alloy constraints for modeling and validation of business rules.

Representation of Business Rules in Alloy
By their definition, business rules are intended to assert business structure or to control or influence the behavior of the business [22]. Structural rules define the business information model. Whereas, a behavioral rule is about how the business reacts to business events. They are specified when something happens at the boundaries of the system [10]. In this approach we deal with both kind of business rules.

According to the rule classification from [22],[10], we distinguish two categories of rules: structural business rules and behavioral business rules. We also distinguish between behavioral rules that have a global scope (represent system invatiants) and those that have limited scope (must hold for a given process, activity or context). We use *Alloy facts* to specify the rules that must hold for entire model (i.e. structural business rules and behavioral rules that are system invariants). For example, *"Every customer record must contain at least one valid billing address."* We use Alloy predicates to model the business rules with a clear scope or context, for example, *"If an order request from a new customer is received, this customer has to be registered in the enterprise system."*

Whereas some behavioral rules can be seen as restrictions or *prohibitions*, thus modeled with Alloy facts and predicates, other business rules have a different nature, for example, *"Strategic customers must always be able to submit the order with GR"*. This business rule is not a restriction, but a *necessity* - a property that has to be ensured or provided despite of any other conditions.[2] We model this type of business rules using *Alloy assertions*.

4 Alloy Specification for Order Processing

4.1 Order Processing: Data Structure

The data structure for the order processing is modeled using Alloy signatures as illustrated below.

```
abstract sig GR {
   orderConfirmedSet: set Order,
   orderDeliveredSet: set Order,
   orderPaidSet: set Order,
   partSet: set Part,
   customerSet: set Customer
}
```

```
one sig GR_pre extends GR {
   orderRequest: one OrderRequest
}
one sig GR_post extends GR {}
```

[2] This distinction has been already proposed in [15], where two modal operators are defined: necessity (with its negation: possibility) and obligation (with its negation: prohibition).

Alloy signatures (**sig**) can be abstract or concrete, can have explicit cardinalities and can contain one or multiple fields. Each field indicates a relation to a corresponding object type and can be considered as an analogy of attributes in object-oriented (OO) languages. For the order processing example, we specify a system - GR - as an Alloy signature illustrated above, characterized by the following fields:

partSet - the set representing all parts (watch components) that can be ordered;
customerSet - the set of customers registered in the GR information system;
orderConfirmedSet - the set of orderes created and confirmed in GR;
orderDeliveredSet - the set of orders (subset of created and confirmed orders) delivered to their customers;
orderPaidSet - the set of orders (subset of created and confirmed orders) paid by the customers.

Similarly to [23], we adapt the state-oriented perspective and specify the execution of order processing in terms of a *state transition:* we define a **pre-state** - GR_pre - that describes the state of a system (GR) before the order processing has been performed and the **post-state** - GR_post - that describes the condition that must hold for the system upon the activity termination.

Once the data structure is defined, we specify how the order processing will be executed (behavior).

4.2 Order Processing: Modeling Behavior

Order processing and its three component processes are modeled as Alloy predicates. These predicates specify a transition between *GR_pre* and *GR_post* states. The proposed specifications are represented as *"black box"*: they do not show *how* the corresponding processes are executed but only the *final result* of their execution visible in GR (i.e. how the GR attributes *orderConfirmedSet*, *orderPaidSet* and *orderDeliveredSet* will change upon the process termination). For example, the *orderCreation* predicate (lines 1-10) declares that *the new order must be created and added to the orderConfirmedSet in the post-state GR_post (i.e. upon the order creation termination).* Along these lines order delivery and payment are specified (lines 11-18).

The *orderProcessing* predicate (lines 19-20) specifies that upon the order processing termination, three processes (order creation, order delivery and order payment) must be accomplished. In Alloy, this corresponds to a logical conjunction of *orderCreation*, *orderPayment* and *orderDelivery* predicates.

4.3 Order Processing: Business Rules

To complete our model of order processing from Section 3.1, we model the following business rules in Alloy: *BR1.1, BR1.2, BR2.1, BR3.1* and *BR4.7.*

The business rules *BR1.1* and *BR1.2* have an explicit scope - the order creation process. According to our BR taxonomy presented in 3.2, these rules are modeled with Alloy predicates *customerExists* and *partExists*:

```
//BR 1.1:  A customer order can be created only for the customers registered in EIS.
pred customerExists{
    one c: Customer|
    (c.name = OrderRequest.name)
    and (c.address = OrderRequest.address)
    and (c in GR_pre.customerSet) }
//BR 1.2: A customer order can be created only for the parts existing in the product catalog.
pred partExists{
    one p: Part|
    (p.partID = OrderRequest.requestedPartID)
    and (p.partInfo = OrderRequest.partInfo)
    and (p in GR_pre.partSet)}
//BR... <other BR for Order creation>
pred orderCreation {
    customerExists and partExists and ... =>   .... }
```

```
pred orderCreation {
    customerExists and partExists and ... =>
    (one aNewOrder: Order| one aCustomer: Customer| one aPart: Part |
    aPart =
    findPartByPartID[GR_pre.orderRequest.requestedPartID,GR_pre.partSet]
    and aCustomer=
    findCustomerByName[GR_pre.orderRequest.name,GR_pre.customerSet]
    and aNewOrder=createOrder[aPart, aCustomer] and
    GR_post.orderConfirmedSet=GR_pre.orderConfirmedSet+ aNewOrder)
    else GR_post.orderConfirmedSet=GR_pre.orderConfirmedSet }
pred orderPayment {
    one aCustomer: Customer | one o: Order | (o in aCustomer.oHistory)
    and !(o in GR_pre.orderPaidSet)
    and GR_post.orderPaidSet=GR_pre.orderPaidSet +  o }
pred orderDelivery {
    one aCustomer: Customer | one o: Order | (o in aCustomer.oHistory)
    and !(o in GR_pre.orderDeliveredSet)
    and GR_post.orderDeliveredSet=GR_pre.orderDeliveredSet +  o }
pred orderProcessing{
    orderCreation and orderPayment and orderDelivery }
```

BR1.1 states that before we start order creation (pre-state), the customer (order requestor) has to be registered in the *customerSet*. Along these lines, *BR1.2* states that for order creation requested part should exist in the *partSet*. This predicate is called within its scope - *orderCreation*. As a result, a new order will be created upon *orderCreation* only if the specified rules are respected.

The business rules *BR2.1* and *BR3.1* must hold for the entire model of order processing. They are modeled with Alloy facts[3] *eventuallyDelivered* and *eventuallyPaid*:

```
//BR2.1: Every confirmed customer order must be eventually delivered to the customer.
fact eventuallyDelivered {
    all o: Order |
    orderDelivery and (o in GR_pre.orderConfirmedSet)
    => (o in GR_post.orderDeliveredSet }
//BR3.1: Every confirmed customer order must be eventually paid by the customer
fact eventuallyPaid {
    all o: Order |
    orderPayment and (o in GR_pre.orderConfirmedSet)
    => (o in GR_post.orderDeliveredSet }
```

These facts claim that when delivery (payment) of orders is provided, all the existing orders will be eventually delivered (paid) in post-state.

[3] Alternatively, if the scope of the model is larger (i.e. it covers not only Order Processing but other activities of GR), these BR can be modeled with predicates as in the previous example.

With the rule *BR4.7*, we ensure that the strategic customer can always create a new order despite of any other conditions. This is the "necessity" rule according to our taxonomy from 3.2; we model this rule with an Alloy assertion.

```
//BR4.7 Strategic customers must always be able to submit the order with GR.
strategicCustomerCanAlwaysOrder: check{
    all c: StrategicCustomer |
    (orderCreation and (c.name = OrderRequest.name) and
    (c.address = OrderRequest.address) and partExists) =>
    (one o: Order |
        o.ocCustomer = c and
         o.ocPart.partID = OrderRequest.requestedPartID and
         (o.ocPart.partInfo = OrderRequest.partInfo) and
          !(o in GR_pre.orderConfirmedSet) and
            (o in GR_post.orderConfirmedSet)) }
```

The assertion claims that whenever there is an order request for the strategic customer for the existing part, the new order is created in *EIS*. This assertion can be checked for our model and must be always valid, which means that this business rule is respected in the system. In the case we get counterexamples, the rule is not respected and we need to revise the model.

5 Business Rules Discovery and Validation with Alloy Analyzer

In this section, we demonstrate how the Alloy Analyzer tool can assist in interactive discovery and validation of business rules that have been missing/omitted/implicit in the initial business specification of Order Processing. We terminate this section by generalizing our approach in a form of four steps an analyst needs to accomplish.

5.1 Order Processing: Model Simulation and Business Rules Discovery

Our approach to BR discovery is based on simulation. The objective of model simulation is two-fold: first, to check our model for consistency (absence of contradictory constraints in business rules); and second, to test the random set of model instances generated by Alloy Analyzer. These instances, in our case, represent the scenarios of order processing enabled by our created model.

The Alloy Analyzer generates the model instances in a form of visual diagrams. Examining these diagrams, an analyst identifies the scenarios indicating implicit, missing, or incorrect business rules. We call this phase a business rule discovery.

Simulating the Alloy model for order processing, we investigate how the status of a customer orders changes during the order processing activity. For a given order, this status can be identified by analyzing the *orderConfirmedSet*, *orderDeliveredSet* and *orderPaidSet* of GR. Note that the same order can be in one or multiple sets at a time. For example, if the order is in the *orderPaidSet* - it is paid. Consequently, if the order is not in *orderPaidSet* in pre-state, but is added into *orderPaidSet* in post-state upon termination of a given activity, it means that it has been paid.[4]

Simulating our model, we find the instances where the customer can make a new order, while some of his previous orders (order history) that have been already delivered remain "unpaid". This scenario is illustrated in Fig. 1. The scenario shows a regular customer (parallelogram on top) creating an order. This customer is associated with 2 orders: *Order0* and *Order1* (black rectangles). *Order 0* is a newly placed order (not in the *orderConfirmedSet* in pre-state); *Order 1* is an old order that is delivered but unpaid (the status is indicated in the bottom of the corresponding rectangle). In the post-state, the new order *Order0* is confirmed (added into *orderConfirmedSet*) and placed in the oder history, meaning that it was accepted by the system.

The presence of this scenario need to be analyzed by a domain specialist as it potentially can bring to a company a lot of unpaid orders and short or long-term loses. The domain specialist decides whether he needs to define new BRs to restrict this behavior and to protect the interest of the company.

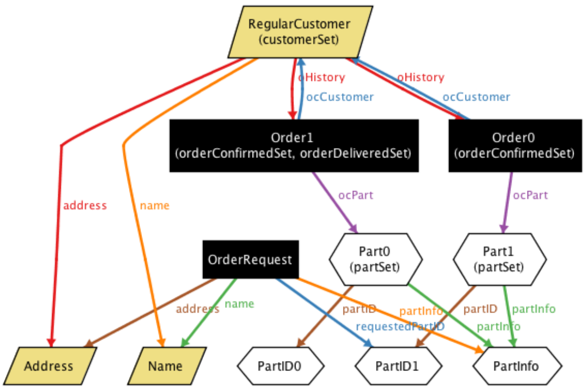


Fig. 1. Order Accepted with Unpaid Orders in the History

5.2 Order Processing: Model Simulation and Business Rules Analysis

We provide a domain specialist with an instant feedback, helping him to reason about the existing business rules, to interactively discover new or implicit business rules and, eventually, to improve their enforcement. Once the new rule is specified by the domain specialist, it can be translated to Alloy for further model analysis. The new business rule covering this business case is:

– BR4.8. New order from a customer can be accepted only if all delivered orders in the customer's order history are paid.

[4] The statuses cannot be canceled, i.e. once the order is paid, it cannot be "unpaid", etc.

This business rule has an explicit context - order creation. We model it with Alloy predicate *customerMustPayDeliveredOrdersBeforeNewOrder*.

```
pred customerMustPayDeliveredOrdersBeforeNewOrder{
    all c: Customer | all o:Order |
    (c.name = OrderRequest.name and c.address = OrderRequest.address and
    o in c.oHistory and !(o in GR_pre.orderDeliveredSet))
    => (o in GR_pre.orderPaidSet)}
```

This predicate claims that, for order creation, all orders from customer's history that are already delivered have to be paid. We modify the *orderCreation* predicate to add this BR.

```
pred orderCreation{
    customerExists and partExists and
    customerMustPayDeliveredOrdersBeforeNewOrder ... =>     .... }
```

As a result, simulating the model, we observe that there are no cases where the order is created and unpaid delivered orders exist for a given customer. However, when we check the validity of the business rule for strategic customer, by checking the assertion *strategicCustomerCanAlwaysOrder*, we receive the result that it is not valid. The counterexamples show the cases where this rule is not respected - no new orders are added in *orderConfirmedSet* for the given order request (Fig. 2). In this example, we see that the new order is not created for the strategic customer, invalidating *BR4.7*. As strategic customers are of crucial importance for GR, they should bypass the new rule and be able to order, even if they have some unpaid orders. In order to resolve current conflict, GR domain specialist modifies the rule 4.8:

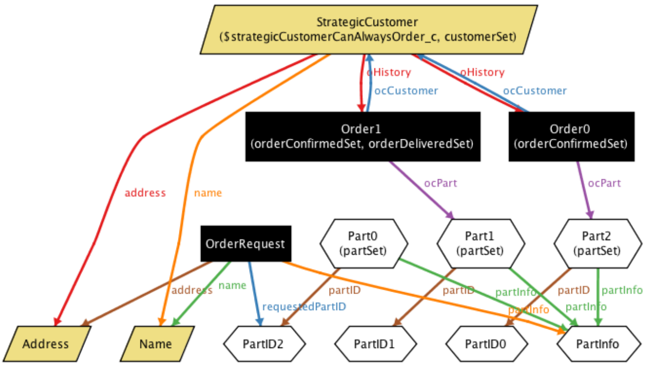


Fig. 2. Example - New Order Not Created For Strategic Customers Because of Unpaid Past Orders

– BR4.8 New order from regular customer can be accepted only if all delivered orders in the customer's order history are paid.

```
pred regularCustMustPayDeliveredOrdersBeforeNewOrder{
    all c: RegularCustomer | all o:Order |
    (c.name = OrderRequest.name and
    c.address = OrderRequest.address and
    o in c.oHistory and
    !(o in GR_pre.orderDeliveredSet))
    => (o in GR_pre.orderPaidSet)}
pred orderCreation {
    customerExists and partExists and customerMustPayDeliveredOrdersBeforeNewOrder ...=> ....}
```

If we check now the assertion *strategicCustomerCanAlwaysOrder*, we do not obtain any counterexamples, which means that the rule is valid again. If we run the order creation process, we can obtain the instances showing that the strategic customer can order even with unpaid orders in his history. Alternatively, the domain specialist can provide the rules for strategic customers, which limit the number of unpaid orders or their total amount.

5.3 Interactive BR Discovery Process: Four Steps

We generalize our approach and define the four steps that a business analyst, business rules designer and/or domain specialist can accomplish in order to systematically discover the business rules and to analyze the issues related to them.

1. The business analyst specifies the BR in a natural language;
2. The designer classifies the BR according to their scope and nature (see Section 3.2) and translates them to Alloy specification language. He also models the whole system of interest (a process, an activity etc) in Alloy, so that he can detect how the business rule influence the system behavior.
3. The designer simulates the model with Alloy Analyzer tool and examines the model instances. [5] Model instances reveal the issues with the existing business rules and indicate the missing or implicit rules. "Necessity" business rules can be validated or invalidated by checking corresponding Alloy assertions.
4. Once a new business rule is discovered, the business analyst specifies this rule in a natural language. Then this rule is added to the Alloy model for further simulations.

6 Conclusion

In this paper, we have presented an interactive, simulation-driven approach for business rule discovery. Our approach is based on formal model checking with the Alloy Analyzer tool. We specify business rules as constraints in Alloy: three types of constraints (facts, predicates and assertions) can be used depending on the type of business rule (necessity/possibility, obligation/restriction) and on their scope.

[5] If no such instances are generated - this indicates the presence of contradictory rules in the model.

We have illustrated our approach with the example of order processing. We demonstrate how the order processing activity and the business rules for it can be modeled in Alloy. We also illustrated how the Alloy Analyzer tool can assist in interactive discovery and validation of business rules. Though formal methods are rarely considered in business, our findings demonstrate that, thanks to their rigor, these methods can support companies, enabling more systematic discovery, analysis and validation of their business rules.

The next step in our approach is to make the language for business rules specification closer to business analyst. One way to do this is to use Attempto Controlled English (ACE) [24], a controlled natural language, i.e. a rich subset of standard English designed to serve as knowledge representation language. This would enable analysts to express the business rules with rigor and in terms of their respective application domain.

References

1. Boyer, J., Mili, H.: Agile Business Rule Development: Process, Architecture, and JRules Examples. Springer (2011)
2. Jackson, D.: Alloy Analyzer tool (2013), `http://alloy.mit.edu/alloy/`
3. Morgan, T.: Business rules and information systems: aligning IT with business goals. Addison-Wesley Professional (2002)
4. ILOG, I (2013), `http://www-01.ibm.com/software/websphere/ilog/`
5. Advisor, F.B. (2013), `http://www.fico.com/`
6. Pegasystems (2013), `http://www.pega.com/`
7. Berstel-Da Silva, B.: Verification of business rules programs (2012)
8. Nagl, C., Rosenberg, F., Dustdar, S.: Vidre–a distributed service-oriented br engine based on ruleml. In: 10th IEEE International, EDOC 2006, pp. 35–44. IEEE (2006)
9. Orriëns, B., Yang, J., Papazoglou, M.P.: A framework for business rule driven service composition. In: Benatallah, B., Shan, M.-C. (eds.) TES 2003. LNCS, vol. 2819, pp. 14–27. Springer, Heidelberg (2003)
10. OMG: OMG: Semantics Of Business Vocabulary And Business Rules (SBVR) - Version 1.0. OMG Document Number: formal/2008-01-02 (2008)
11. Nelson, M.L., Rariden, R.L., Sen, R.: A lifecycle approach towards business rules management. In: Proceedings of the 41st Annual Hawaii International Conference on System Sciences, pp. 113–113. IEEE (2008)
12. Ross, R.G.: Principles of the BR approach. Addison-Wesley Professional (2003)
13. Halpin, T.A., Morgan, A.J., Morgan, T.: Information modeling and relational databases. Morgan Kaufmann (2008)
14. Dietz, J.L.: Enterprise ontology: theory and methodology. Springer (2006)
15. Dietz, J.L.: On the nature of brs. Advances in Enterprise Engineering I, pp. 1–15 (2008)
16. Jackson, D.: Software Abstractions- Logic, Language and Analysis. MIT Press (2011)
17. GR: Generale ressorts site, `http://www.generaleressorts.com/` (2013)
18. Jackson, D., Schechter, I., Shlyakhter, I.: ALCOA: The Alloy constraint analyzer. In: Proceedings of the 22nd ICSE, Limerick, Ireland (June 2000)
19. Rychkova, I.: Formal Semantics for Refinement Verification of Enterprise Models. PhD thesis, EPFL (2008)

20. OMG: OMG: Object Constraint Language - Version 2.2. OMG Document Number: formal/2010-02-01 (2010)
21. Bajic-Bizumic, B., Wegmann, I.R., Towards, A.: a invariant-based service design process. Technical report, EPFL (2013)
22. Hay, D., Healy, K.A.: Defining brs -what are they really. Final Report (2000)
23. Andersson, T., Bider, I., Svensson, R.: Aligning people to business processes experience report. Software Process: Improvement and Practice 10(4), 403–413 (2005)
24. ACE: Attempto Controll English (2013), http://attempto.ifi.uzh.ch/site/

Application of IT Management Frameworks in Higher Education Institutions

Martin H. Knahl

Faculty of Business Information Systems,
University of Applied Sciences Furtwangen, Furtwangen, Germany
knahl@hs-furtwangen.de

Abstract. IT Management best-practice process frameworks for IT Service Management and IT Governance have been applied by many organizations to structure and improve operational IT management and IT governance. It further facilitates customer centric, cost efficient and compliant IT Service provisioning. IT Management has undergone various evolutionary stages and changes over recent years due to the evolving maturity of best-practice recommendations and evolving IT requirements. This paper refers to the current practices and standards in the areas of IT Service Management and IT Governance. A case study identifies and outlines implications for the Higher Education sector. The practical adoption of these frameworks in higher education institutions suggests that the domain has advanced and matured in recent years. However further research and adjustments are required to further facilitate the adoption of the core ITSM and IT Governance principles.

Keywords: IT Service Management, IT Governance, ITIL, CobiT.

1 Introduction

Service Science for IT and related technologies holds high importance in the current business world with organizations recognizing IT not only as a cost center but also as a crucial factor in enabling the success of an organization through IT Alignment. This manifests itself in areas such as IT Governance and IT Service Management (ITSM) [1] [2]. Organizations aim to improve the IT Services delivery by adopting IT service management processes and transforming their organization from a provider of IT focused functions into a customer-centric organization meeting customer requirements [3]. Such an IT context requires the organization to effectively identify required IT Services, plan the development of new services, support operational services and continuously redefine service provisioning. Thus IT Management should cover the entire lifecycle of IT Services. ITSM and IT Governance aid in this scenario by offering frameworks that act as models for organizations in implementing the best practices and improvising their service offerings. ITSM aligns the business and IT of an organization by directing IT efforts to achieve business demands and achieving customer satisfaction by ensuring the quality of the services. The alignment of IT usage and business needs in an organization in an appropriate manner can be achieved with

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 124–133, 2013.

the help of IT governance frameworks and principles [4]. IT governance defines the structure of relationships and processes to develop, direct and control the entirety of the IT resources in order to achieve the goals of an of an organization. IT governance thus aims to ensure control over the implementation of IT strategy thereby ensuring the proper alignment of the IT and the Business functions.

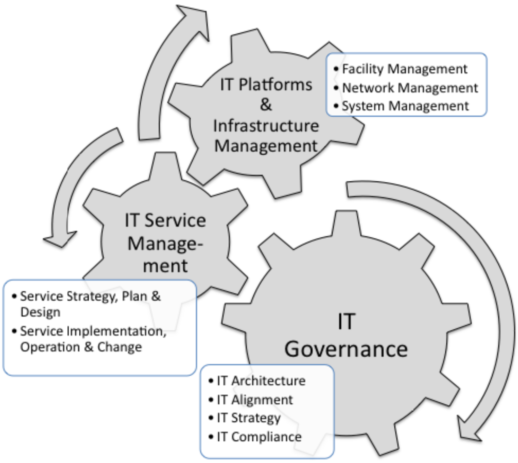


Fig. 1. IT Management Domains

IT Governance is a subset domain of Corporate Governance focused on IT business alignment, performance, risk and compliance. The rising interest in IT Governance is partly due to compliance requirements and legislation such as the Sarbanes-Oxley Corporate Governance legislation that force organizations to comply with a more or less well defined set of rules and regulations as well as the acknowledgment that a large number of IT initiatives fail to deliver the desired results and lack business alignment. Given the implicit requirements to embed IT Governance in the Corporate Governance framework it becomes clear that this topic is relevant to most organizations. As explained by R. Nolan *"a board needs to understand the overall architecture of its company's IT applications portfolio … The board must ensure that management knows what information resources are out there, what condition they are in, and what role they play in generating revenue"* [5]. IT Governance envisages a system where the various stakeholders (e.g. management, IT department, internal customers) become part of the decision making process to prevent IT from independently making poor decisions (e.g. that a system does not behave or perform as expected).

IT Service Management (ITSM) is a discipline for managing IT Services that is centered on the customer's perspective of the contribution of IT to the business (i.e. the role of IT Services to support and facilitate the business) and, in contrast to IT Governance, is mainly concerned with the operational aspects of IT Services. ITSM stands in deliberate contrast to technology-centered approaches (Facility, Network, System and Application Management) that nevertheless provide the basis for meaningful and effective ITSM and IT Governance [6].

IT Service Management and IT Governance has been high on the agenda for many organizations over recent years [2]. Various best-practice frameworks and standards have been adopted by many organizations including higher-education in an effort to improve the quality and effectiveness of the IT Services and the efficiency of IT Service delivery, to develop a customer oriented approach to IT Service provision and to provide a meaningful basis for required IT Governance mechanisms.

This paper is organized into four sections. The first section provides an introduction to ITSM and IT Governance and a motivation for the current the research. The second section highlights the relevant best practice frameworks and standards, their adoption in research and industry and analysis of the frameworks. The third section consists of a synthesis of conclusions drawn from the research and analyses possibilities on improving and adopting these frameworks for the future. The final section provides the conclusions and future outlook.

2 IT Management Frameworks and Standards

“To the majority of computer scientists, whether in academia or industry, the term ‘services’ is associated with Web services and service-oriented architectures. However, there is a broader story to be told of the remarkable growth of the service sector, which has come to dominate economic activity in most advanced economies over the last 50 years. The opportunity to innovate in services, to realize business and societal value from knowledge about service, to research, develop, and deliver new information services and business services, has never been greater. The challenges are both the multidisciplinary nature of service innovation, which combines business, technology, social-organizational, and demand innovation as well as the lack of formal representations of service systems“ [7].

Sporer & Riecken refer to the relevance of service thinking and the adoption of IT service concepts such as Service-Oriented Architectures [7]. There are several standards, and best practice frameworks to define, plan, operate and manage IT Services that aim to embed the concept of service thinking into an organization [2]. The most relevant and widely used best practice frameworks and standards in the context of IT Management are ITIL, ISO 20000, CobiT and ISO/IC 27002 [2-4]. CobiT focuses on how to deliver information to satisfy business needs, ITIL is concentrated on the delivery of IT Services (i.e. a specific business need), ISO 20000 provides a standard broadly based on ITIL for IT Service Management and ISO/IEC 27002 provides a standard for information security.

IT Infrastructure Library (ITIL) is a widely cited and adopted de-facto standard that often provides the basis of IT Service delivery and support processes in industry. It is widely accepted across the world in public and private sectors as a reference framework and practical basis for IT Service Management [2]. It was first introduced and distributed by the Office of Government Commerce (OGC) in the UK in the 1980s in the light of growing IT complexity and costs to facilitate structured IT service delivery. ITIL aims to provide process definitions and descriptions for the entire IT function and service delivery of any given organization with a focus on IT service management. The early versions where predominantly concerned with operational support processes whereas the current version of ITIL aims to cover all stages of IT Services. The current version ITILv3 and its subsequent refinement ITIL2011 is based on a lifecycle model that consists of five key stages (and each stage is published as a separate ITIL book): Service Strategy, Service Design, Service Transition, Service Operation and Continual Service Improvement. The objective of ITIL processes is to define, evaluate and improve the quality of services and service delivery. It thus helps to make the services available to the customers to support the business function, thereby improving and providing high quality service management.

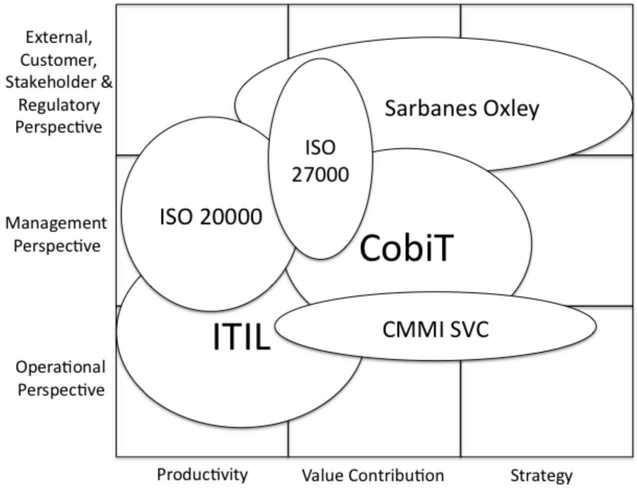


Fig. 2. IT Management Reference Models & Frameworks

Control Objectives of Information and related Technology (COBIT) is a set of best practices for Information Technology management created by the Information Systems Audit and Control Association (ISACA) and the IT Governance Institute (ITGI). COBIT is the internationally accepted framework for IT Governance and Control [4]. It was first introduced in 1992, the current version CobiTv5.

COBIT aims to provide a comprehensive approach to all IT Governance activities. It facilitates a better understanding of the IT systems and a control and security framework to protect the company's assets. COBIT is a widely recognized and accepted IT governance framework [4]. It defines management domains that consist of key IT Governance processes or control objectives, which are broken into detailed IT controls. The management domains roughly represent the lifecycle of the overall IT Governance process: Plan and Organize, Acquire and Implement, Deliver and Support, and Monitor and Evaluate. For each of the processes COBIT defines performance goals and metrics, Key Performance Indicators (KPIs) and maturity levels in assisting benchmarking and decision making for process improvements. It provides additional guidance for organizations such as RACI charts identifying who is Responsible, Accountable, Consulted and or Informed for specific IT processes.

ISO/IEC 27002 is an Information Security Management System (ISMS) standard and is often referred to as a basis or a code of practice for Information Security Management [8]. It lists security control objectives and recommends a range of specific security controls. The ISO 27000 family can be treated more as information security norm rather than a comprehensive IT Governance tool. ISO/IEC 27002 ensures overall security at all levels in an organization. This standard can further be supplemented by national guidelines or recommendations such as the BSI Grundschutz Katalog (Baseline Security Catalogue published by the German Federal Agency for Security in Information Technology), a comprehensive baseline security standard that is evolving towards ISO 27000.

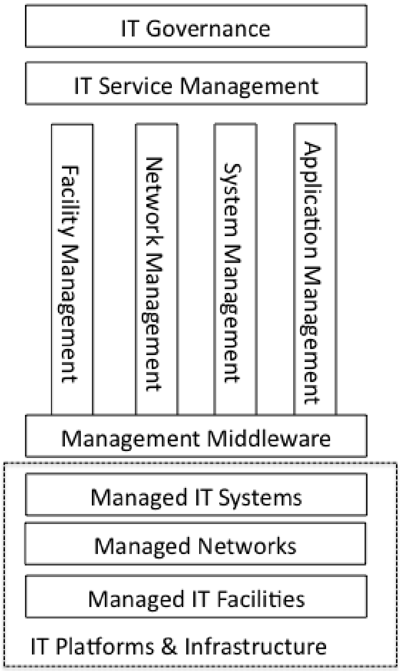


Fig. 3. IT Management Layers

The different management domains can be grouped into a layered model. The Managed IT Platforms and Infrastructure (e.g. comparable to Managed Objects in the context of SNMP) act as the basis for the management of the entire IT infrastructure that provides the basis for IT Service delivery. The aim of a management platform is to integrate the various areas (i.e. Facility, Network, System and Applications Management) of IT infrastructure management that forms the basis for the IT Service Management.

From a commercial perspective, many organizations have adopted ITIL as a basis for their consultancy, service or software offerings. Furthermore there are additional frameworks such as Hewlett Packard's IT Service Management Reference Model

(ITSM), Microsoft Operations Framework (MOF) and IBM's Systems Management Solution Lifecycle (SMSL). However, after originally specifying proprietary models, the vendors now position and align their frameworks upon the ITIL best practice framework, given its widespread acceptance and adoption in the market [9].

3 IT Management in Higher Education

In recent years, the demand and complexity of IT Services in Higher Education has been growing continuously whilst budgets for IT operations and research infrastructure are becoming ever more difficult to sustain. Furthermore, institutions need to consider how to incorporate continuously changing governance structures and how requirements can be aligned with IT Governance especially in the context of ad hoc and unplanned IT engagement [10].

However, according to the market research organization Gartner interest and adoption of COBIT in Higher-Education is rather limited, given its main adoption as an industry audit and benchmarking framework (e. g. to facilitate IT compliance requirements) and thus its commercial, non-academic background. This explains Gartner's Benefit Rating of *Moderate*, Maturity status of *Emerging* and a market penetration of less than 1% of target audience [11]. Typically CobiT is used as an extension or add-on to existing ITIL activities. Given the fact that ITIL itself is gaining wider attention a limited growth in CobiT interest and adoption can be expected. According to Gartner, current interest and adoption of ITIL is somewhat more widespread, with a market penetration of 5% to 20%, a benefit rating of Moderate and a Maturity status of Adolescent [11]. This confirms a somewhat limited acceptance of IT Management best practice frameworks at the moment and the initial adoption of ITIL to improve the IT Service provision for the customers as the crucial starting motivation.

Opportunities	Threats
Aligned and robust IT Services, facilitates Portfolio Management for IT Services	Complicated Processes & administrative overhead
Defined responsibilities, performance incentives for IT Staff	Staffing requirements & resistance (e.g. to change)
Customer centred IT, measurable & repeatable processes	Dependency upon ITIL Services, Tools and Frameworks
Transparent IT Operating Model, facilitates cost management & efficient IT resources utilisation	IT Governance, ITSM or organisational Discontinuity

Fig. 4. Opportunities & Threats for ITSM in Higher Education Institutions

To assess the impact of ITSM, the implications of traditional versus modern IT Service provision have been distilled and a comparison of opportunities and threats analysis has been conducted as part of the research (e.g. interviews and questionnaires with IT Services stakeholders). The results are illustrated in Figure 4. This could further be extended by a SWOT analysis, a planning method used to evaluate the Strengths, Weaknesses, Opportunities, and Threats involved in a business decision, strategy or project [12]. It involves specifying the objective of the project (provision of IT Services that meet quality criteria and requirements) and identifying the internal and external factors that are favorable and unfavorable to achieve that objective. It is interesting to note that costing aspects and the notion of change feature prominently in the various categories.

Breiter [13] and Wild [14] further identify distinct requirements for IT Service Management in the Context of Higher Education organizations. An analysis of the requirements in the context of Higher Education organizations highlights the changing nature of IT Service provisioning. The key requirements have been mapped to the areas IT Governance, ITSM and Facility, Network and System Management in Figure 5.

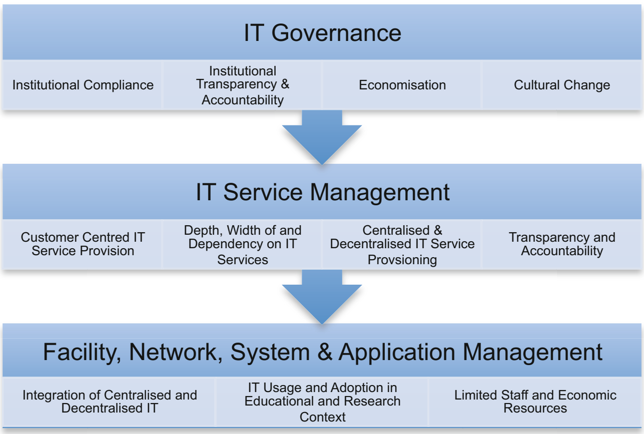


Fig. 5. IT Management Levels

The results of the analysis of the author’s organization, derived from the opportunities and threats of ITSM, together with studies on the implementation of IT governance and ITSM suggest that there are a number of benefits from the adoption of frameworks such as ITIL or CobiT [13-17].

Furthermore, the appropriate IT Governance processes need to be identified and selected. In the case study at the author’s institution, a set of interviews has been conducted prior to the analysis of IT goals with the main aim to identify driving IT goals to identify a base of core IT Governance challenges. Subsequent interviews after the analysis of the IT goals, mainly with IT orientated staff, were performed to complete the identification of IT goals.

It became evident that employees with different roles have widely differing interpretations and classifications regarding the governance of IT processes and the IT function. It further became clear that the relevance of some IT functions by general IT users, IT employees and the IT Officer of the Faculty varied widely, in some cases even contradicting each other (e.g. general IT staff stated that an organizational framework providing a defined Service Level Process as outlined by DS1 is of limited use whereas the IT Officer identified it as a core Process). That represents an interesting observation that IT processes are generally considered and evaluated from different perspectives: the evaluation of the IT process from the perspective of the executing co-workers and the management point of view that emphasizes the planning and structuring aspects of a process. Thus the definition of Critical Success Factors together with metrics in the form of Key Performance Indicators (e.g. number of raised incidents, average workaround time for incidents) can provide a starting point for the different stakeholders.

According to the results of an employee survey Service Level Agreements and Operating Level Agreements are of limited importance for general IT stakeholders. The background for this assessment is the operational ad-hoc practice of providing services to customers (students) and the varying and unspecific definition of performance to be provided with this service. The view of the employee is further enforced by the fact that a supply agreement in a public institution like a university is difficult to implement, especially since the agreement is not contractually or legally binding. However for the IT officer and general IT management, the introduction of SLAs and OLAs are important, since a continuous availability of IT services according to requirements and the definition of service levels will ultimately ensure the end-user satisfaction and success of the institution.

While the higher education sector generally has to cope with low staffing levels, this should not become a deterrent in adopting industry best-practices. It is further agreed upon that it is important to evaluate the strengths and weaknesses as a basis fort he selection process (e.g. selection of ITIL processes). Given the typically decentralized nature and the running of similar, overlapping IT Services (e.g. multiple learning and teaching management platforms) leads to duplication of ICT staff without necessarily increasing the quality of the IT Services. Thus a consolidation of IT Services may lead to the availability of additional staffing resources. This generally requires a sound communication between central IT and distributed IT staff to enable acceptance of IT standards. Furthermore improved communication between IT (IT Governance and ITSM) and management (Corporate Governance) will strengthen the position of the IT department as a service provider, rather than a technology provider adding costs to the organisation.

Both CobiT and ITIL are relevant best-practice frameworks and overlap in a number of core areas and processes. One important difference between COBIT and ITIL is the availability associated with the approaches (e.g. with regard to cost of documentation). The core COBIT specification and related documentation is available freely, e.g. from the Information Systems Audit and Control Association (ISACA) website. On the other hand ITIL specifications and a large body of additional documentation can become costly for an organisation (e.g. is not freely available).

Another challenge can be found in the effective planning, communication and management of IT Services from the provider and the customers. The difference of perceptions (e.g. with regards to the quality of an IT Service) across an institution of higher education may thus lead to difficulties in managing the relation of students and staff. In a Higher-Education context, student dissatisfaction with IT Services (e.g. at the faculty level) is reflected in general in internal surveys and national surveys that provide the basis for rankings (e.g. CHE ranking in Germany). A consolidation of services (e.g. university wide teaching and learning platform, centralised service desk) facilitates the maintenance of standards and perceptions across an organisation.

4 Conclusions and Outlook

The changing nature of IT Service provisioning in the context of higher education together with ever increasing dependency on IT necessitates the adoption of best practice and research frameworks to ensure alignment, quality and cost efficiency. Thus, “Best Practice” reference models such as ITIL or CobiT and related standards can provide a basis for implementation.

Future work must further concentrate on the mapping of IT Management theory and practice. The realities of higher education organizations for the provision of IT Services using and integrated management architecture and process model remain challenging. However the study has demonstrated that the adoption and integration of management tools with “Best-Practice” Frameworks have the potential to facilitate and enhance IT Management.

References

1. Chesbrough, H., Spohrer, J.: A research manifesto for services science. Communications of the ACM 49(7) (2006)
2. Galup, S.D., Dattero, R., Quan, J.J., Conger, S.: An overview of IT service management. Communications of the ACM 52(5) (2009)
3. Winniford, M.A., Conger, S., Erickson-Harris, L.: Confusion in the Ranks: IT Service Management Practice and Terminology. Information Systems Management 26(2), 153–163 (2009)
4. Johannsen, W., Goeken, M.: Referenzmodelle für IT-Governance. dpunkt. verlag GmbH, Heidelberg (2007)
5. Nolan, R., McFarlan, F.W.: Information Technology and the Board of Directors. Harvard Business Review. Harvard University, Boston (2005)
6. Pavlou, G.: On the Evolution of Management Approaches, Frameworks and Protocols: A Historical Perspective. Journal of Network and Systems Management 15(4), 425–445 (2007)
7. Spohrer, J., Riecken, D.: Special section on services science (Introduction). Comm. of the ACM 49(7), 30–32 (2006)
8. Wang, C.-H., Tsai, D.-R.: Integrated installing ISO 9000 and ISO 27000 management systems on an organization. In: 43rd Annual 2009 International Carnahan Conference on Security Technology, pp. 265–267 (2009)
9. Rüter, A., Schröder, J., Göldner, A., Niebuhr, J.: IT-Governance in der Praxis. Xpert.Press. Springer, Heidelberg (2010)
10. Krueger, D.: Decentralized IT Governance and Policy in Higher Education. Educause Center of Applied Research Research Bulletin 2009(5) (2009)
11. Lowendahl, J.-M.: Hype Cycle for Education, 2011. Gartner Industry Research (July 29, 2011) (2011), `http://www.gartner.com/DisplayDocument?id=1755718`

12. Houben, G., Lenie, K., Vanhoof, K.: A knowledge-based SWOT-analysis system as an instrument for strategic planning in small and medium sized enterprises. Decision Support Systems 26(2), 125–135 (1999)
13. Breiter, A.: IT Service Management an Hochschulen im Spannungsfeld zwischen Zentrale und Dezentrale. In: Campus Innovation, Hamburg, November 27-28 (2009)
14. Wild, P.: IT Service Management an Hochschulen: Aufbau der Supportorganisation an Hochschulen mit Campus Management Systemen. Vdm Verlag Dr. Müller (2008)
15. Bhattacharjya, J., Chang, V.: Adoption and Implementation of IT-Governance – Cases from Australian Higher Eduction. In: Cater-Steel, A. (ed.) Information Technology Governance and Service Management, pp. S.82–S.100. Information Science Reference/IGI, Hershey (2009)
16. Hicks, M., Pervan, G., Perrin, B.: A Case Study of Improving Information Technology Governance in a University Context. In: Pries-Heje, J., Venable, J., Bunker, D., Russo, N.L., DeGross, J.I. (eds.) IFIP WG. IFIP AICT, vol. 318, pp. 89–107. Springer, Heidelberg (2010)
17. Wannemacher, K., Moog, H., Kleimann, B. (publ.): ITIL goes University? Serviceorientiertes IT-Management an Hochschulen Konzepte und erste PraxiserfahrungenHIS: Forum Hochschule (2008)

Using ArchiMate and TOGAF to Understand the Enterprise Architecture and ITIL Relationship

Marco Vicente[1], Nelson Gama[1,2], and Miguel Mira da Silva[1]

[1] Instituto Superior Tecnico, Av Rovisco Pais, 1049-001 Lisboa, Portugal
[2] CINAV-PT Navy Research Center, Escola Naval, Alfeite, 2810-001 Almada
{marco.vicente,nelsongama,mms}@ist.utl.pt

Abstract. Business/IT alignment has become one of the most relevant concerns on organizations. Enterprise Architecture (EA) and ITIL, two distinct governance approaches with different perspectives, have become recently dominant between practitioners. However, parallel EA and ITIL projects can lead to wasted resources and a duplication of costs and efforts. In this paper we propose an EA and ITIL integration using ArchiMate as a common frame of reference. We also want to point out that implementing ITIL is like implementing any other architecture change and demonstrate it by using TOGAF to perform an ITIL implementation on ArchiSurance, a fictitious organization from the well known ArchiMate case study.

Keywords: Enterprise Architecture, ITIL, IT Service Management, business/IT alignment, TOGAF.

1 Introduction

In the last decades, IT has evolved from its traditional orientation of administrative support to a strategic role, turning business/IT alignment into a major concern. In the early nineties, Henderson [1] proposed a strategic alignment model based on two building blocks: strategic fit and functional integration, using business strategy as the driver and IT as the enabler. This model presented several perspectives on how to integrate business and IT domains, using concepts like information systems service organizations and IT governance [1].

Recently, the growing demand on IT led to the improvement of the key concepts related to IT Governance, namely the ones connected to IT alignment with strategic objectives and cost reduction initiatives [2]. From these Governance initiatives, two main approaches have had major relevance: Enterprise Architecture (EA) and IT Service Management (ITSM).

EA is a coherent whole of principles, methods, and models that are used in the design and realization of an enterprise's organizational structure, business processes, information systems, and infrastructure [3]. Therefore, according to EA approaches, organizations usually share several architectures: business, processes, information, application and technology infrastructure [3,4,5].

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 134–145, 2013.

ITSM evolved naturally as services became underpinned in time by the developing technology. In its early years, IT was mainly focused on application development, but as time went by, new technologies meant concentrating on delivering the created applications as a part of a larger service offering, supporting the business itself [6]. ITIL [7] is the *de facto* standard for implementing ITSM [8]. It is a practical, no-nonsense approach to the identification, planning, delivery and support of IT services to the business [9].

Unfortunately, having two different frameworks to approach governance can led to several setbacks. In a time when organizations strive to be efficient and effective, it seems counterintuitive to be wasting resources by having different organizational departments or teams handling both approaches independently.

This paper is part of a wider effort to join both approaches by establishing a specific enterprise architecture for organizations that need to manage IT services. EAs don't focus on specific issues because their goal is to be able to represent every organization. On the contrary, our goal is to narrow it down, and restrict the architecture to organizations that have the management of IT services as an architectural driver, using ITIL principles, methods, processes and concepts to perform IT service management, and general EA principles, methods and models to the design and realization of the remaining organizational structure.

Thus, this paper's goal is to show how ITIL can be approached from an EA point of view, and how this can be used to design ITSM organizations and implement ITIL through EA methods as if it were any other architectural change.

The methodology applied across this paper is Design Science Research, where we develop and validate a proposal to solve our problem [10]. The following sections follow the methodology's steps: "Related Work" covers aims and objectives as the awareness and recognition of a problem from a state of the art review giving us the issues that must be addressed. Later on, "Research Problem", exposes the main problem while offering a tentative idea to how these issues might be addressed. Afterwards, "Proposal" presents a proposal as an attempt to solve the previously described problem. Next, we present a "Demonstration" followed by the "Evaluation" comparing the results with the research questions and to conclude we show our proposal applicability and themes for further work.

2 Related Work

Here we will introduce what is Enterprise Architecture, followed by TOGAF - an EA framework, and ArchiMate - an EA modeling language. Finally we will address ITIL, a best practices model to IT service management.

2.1 Enterprise Architecture

Today, business performance depends on a balanced and integrated design of the enterprise, involving people, their competencies, organizational structures, business processes, IT, finances, products and services, as well as its environment [11].

EA is a coherent set of principles, involving the design and performance of different architectures. It specifies the components and its relationships, which are used to manage and align assets, people, operations and projects to support business goals and strategies [3,12], concerning those properties of an enterprise that are necessary and sufficient to meet its essential requirements [11]. EA is based on a holistic representation of organizations, on views and the ability to map relationships between artifacts and architectures which usually are [3,4,5]: Business, Process, Application, Information, and Technology. The alignment between them allows a coherent blueprint of the organization, which is then used for governance of its processes and systems [13].

2.2 TOGAF

The Open Group Architecture Framework (TOGAF) is a framework for developing an EA [4]. It was developed and is currently maintained as a standard by The Open Group (TOG). The first version of TOGAF, in 1995, was based on the US Department of Defenses Technical Architecture Framework for Information Management (TAFIM) [4,14]. Each version of the standard is developed collaboratively by the members of the TOG Architecture Forum [4,14].

TOGAF provides the methods and tools for assisting in the acceptance, production, use, and maintenance of an EA [4]. It is one of the leading architecture frameworks worldwide, and in its latest version there is increasing reflection on the use of the architecture and its governance [14]. The TOGAF documents focus on EA key concepts and TOGAF Architecture Development Method (ADM), a step by step approach to developing an EA [15].

2.3 ArchiMate

The ArchiMate EA modeling language was developed to provide a uniform representation for architecture descriptions [15,16]. The goal of the ArchiMate project is to provide domain integration through an architecture language and visualization techniques that picture these domains and their relations, providing the architect with instruments that support and improve the architecture process [25]. ArchiMate is now also becoming well known in the international EA community, being today a TOG standard [15].

The domains of business, application and infrastructure are connected by a service orientation paradigm, where each layer exposes functionality in the form of a service to the layer above [16]. Besides this, it also distinguishes between active structure, behavior and passive structure elements, having also another distinction between internal and external system view. On top of this, ArchiMate is a formal visual design language, supports different viewpoints for selected stakeholders and is flexible enough to be easily extended [16].

2.4 ITIL

ITIL was created by the Central Computer and Telecommunications Agency (CCTA), and was first released to the public in the late eighties [14]. ITIL is a common-practice model possessing the character of a branch standard [8]. While the first version was mainly based on experience in data centers running big mainframes, in 2000 a revised version (v2) was launched becoming the worldwide de facto standard for IT Service Management [14].

The next version of ITIL (v3) appeared in 2007 and covers the major weaknesses identified in the previous versions, namely being too focused on technology [2]. Now, instead of focusing on the service itself, the focus lay on this cycle of life, renewal and decommissioning of services, with a greater business-focused perspective [14]. The ITIL Core consists of five publications and each book covers a phase from the Service Lifecycle with various processes which are always described in detail in the book in which they find their key application [17].

3 Research Problem

There have been some attempts to integrate EA and ITIL. In fact, Brown and Winter [18] proposed an EA expansion to integrate ITIL v2 and Service Oriented Architectures (SOA), having EA as a pivotal concept with ITIL regarded for IT operations. Nabiollahi [19] provides a service based framework for EA to meet the ITSM requirements of ITIL, suggesting an EA extension to involve service architecture layer from ITIL Service Design [20]. The development of an architecture model for IT services is proposed, making it a service layer for EA.

Thorn [21] addresses the relation between ITIL and TOGAF, regarding EA as a fundamental concept for organizational engineering, in which ITIL is included as a framework to an operation model for IT delivered services. He argues that both frameworks can be used together by mapping them, TOGAF covers the development of EA, and is involved in the products conception lifecycle whereas ITIL ensures the delivery and management of IT services to users [2,21].

In the same note, Sante [14] addresses the fact that the recent versions of ITIL and TOGAF keep converging to integration. In fact, in ITIL V3 references are made to architectural concepts, hitherto only found in publications on architecture. The same, although to a much lesser extent, applies to TOGAF 8.1.1: references are made to IT management [14]. The author relates the five ITIL books to TOGAFs ADM cycle, showing there are indeed several similarities, but identifying two main differences: a) developing business architecture is part of the TOGAF framework while the scope of ITIL is limited to developing an effective and efficient IT department, whilst developing business architecture is out of scope in ITIL; and b) running IT operations and delivering actual IT services are within the scope of ITIL, while TOGAF does not cover the development and maintenance of a run time environment, neither the way how services are actually produced and delivered [14].

Thus, EA is regarded as a fundamental concept for organizational engineering, and ITSM is regarded as the dominant operations model sufficiently integrated into the former [22]. EA guarantees consistency in building new products or services and addresses business requirements, while ITSM guarantees the consistency of services, through the use of standard processes [22].

All these integration attempts tried to answer a real problem that should not be taken lightly. However, while all these approaches seem to come close to an integration, they don't propose a definitive and holistic solution. In fact, Braun's [18] and Thorn's [21] work is limited to ITIL v2, what makes it outdated, Nabiollahi [19] proposes a service architecture as a new architecture layer, but doesn't clarify the architectures relationships. As for Sante's [14] work, although we agree upon the approaches used and the conclusions reached, the result isn't a unique body of knowledge with EA and ITIL, but two different frameworks linked by a mapping. Moreover, none of these approaches provide models or a formal representation for the proposed solutions. In fact, what we are looking for is a holistic solution, following the EA approach but using ITIL best practices to perform IT service management.

Our research question is then how can we contribute to this discussion and how can ITIL be integrated with EA in a way that would allow ITIL and EA teams to collaborate on organizational change.

4 Proposal

ITIL is often described as a process-based [6] or a process-oriented [17] framework. Although we realize that most of ITIL contents are about describing best practice processes (and the information they use), we believe that limiting ITIL to these only two domains is one of the factors that makes its integration with EA so difficult. That said, we believe that like EA, we should also look at ITIL as a composition of other architectures, linked by a service oriented approach, where functionality is available to the next layer in the form of services.

Additionally, we also realized that it would be harder to integrate two approaches if they didn't speak exactly the same language, so we needed a uniform representation, a common frame of reference. To represent EA we chose ArchiMate as it offers an integrated architectural approach that describes and visualizes the different architecture domains and their underlying relations and dependencies [16]. As for ITIL, it is an English natural language set of documents consisting of several volumes of IT management concepts, processes and methods [8]. Therefore, in the absence of a formal ITIL graphical language and based on our belief that ITIL can be regarded as part of EA, sharing the same domains, components and relationships, we decided to try to model the ITIL meta-model using the language we had already chosen for EA: ArchiMate.

We want however to emphasize that providing a common meta-model of EA and ITIL is a different issue from using a EA oriented language for representing ITIL. In fact, we are proposing the former, so this work isn't the enterprise architecture of ITIL but the modeling of ITIL inside the organization's EA.

Hence, we started by searching through the five ITIL books for concepts that belonged to each of the EA domains. Having them, we built a mapping of these concepts to ArchiMate's metamodel elements based on ITIL and ArchiMate's own definitions. With this concept mapping, we built the ITIL metamodel using ArchiMate concepts for the whole ITIL 26 processes and 4 functions.

At the end, although we had answered the Zachman Framework's "What", "Where", "When", "Who" and "How" questions, we still lacked the "Why".

To answer this last one, we had to extend the "what is done by the system" approach with the "why is the system like this". This why question is answered in terms of organizational objectives and their impact on information systems supporting the organization [23]. Thus, we chose ArchiMate's Motivation Extension which is used to model the motivations, or reasons, that underlie the design or change of some EA. These motivations influence, guide, and constrain the design [16]. Here we followed a similar course of action: we identified ITIL motivational concepts, mapped them to ArchiMate's concepts, and built motivation models for the ITIL 26 processes and 4 functions.

The research for ITIL graphical representations, the concept maps and the description of the method for the models' construction are out of the scope of this article, and were addressed in two other papers, one about ITIL Business Motivation Model [24] and another one that is awaiting publication.

As we have stated, it is quite possible to identify ITIL components and relationships in every EA domain. Thus, if one starts looking at ITIL from this point of view, we begin to realize that by representing and splitting it across EA realms, we can use composition to integrate them by integrating each of its layers. Therefore, our first proposal is: if an organization can be represented by an enterprise architecture, with all its layers, components and relationships, and if that organization has implemented ITIL, then ITIL components and relationships will be a subset (in every layer) of the EA ones.

On the other hand, an architecture model is not just useful to provide insight into the current or future situation; it can also be used to evaluate the transition from 'as is' to 'to be' [3], and there is a strong relationship between implementing a target EA and an ITIL program. These relationships are manifested in terms of People, Process, Business, and Information [26]. Thus, based on our first proposal that ITIL is part of EA, in the sense that if an organization has ITIL, then in every EA layer there will be ITIL elements, then our second proposal is: implementing ITIL on an organization represented by an EA is the same as implementing any other architectural change, so an EA method for the transition from a baseline to a target architecture could be used to implement ITIL.

5 Demonstration

To demonstrate how ITIL fits in EA, we needed to present an organization EA model containing ITIL elements. To achieve this, we decided to use ArchiSurance. The ArchiSurance Case Study is a fictitious example developed to illustrate the use of ArchiMate in the context of TOGAF [27]. It concerns the insurance company ArchiSurance, which has been formed as the merging of three independent companies. It describes the baseline architecture before the merging and then a number of change scenarios. TOGAF ADM is then used to go from that baseline architecture to a target one with ArchiSurance after the merging.

Since this is a running example that is widely used across the ArchiMate community [3,15,16,25,28,29] and on ArchiMate training courses [27] we thought it would fit our demonstration purposes. Moreover, The Open Group "expects it to evolve over time, and encourages its members to add new aspects and views or create new change scenarios, as long as they are consistent with the original case description and models" [27]. That said, we start by pointing out that our models are indeed consistent with the existing ones, since we don't subtract anything but add ITIL instead. In fact, our baseline architecture is the target of the ArchiSurance example. Our premise is that after the merging, ArchiSurance was facing the same problems that several other organizations face when they decide to use ITIL. Thus, we will use the exact same approach that is used on the ArchiSurance scenarios examples: we will use the TOGAF ADM and ArchiMate to represent an architecture change from a baseline ("as-is") of ArchiSurance (after the merging) to a target ("to-be") architecture with the implementation of ITIL Service Operation.

Therefore, in the *Phase A: Architecture Vision* we establish an architecture effort and initiate an iteration of the architecture development cycle by setting its scope, constraints, and goals. Some relevant drivers, assessments and goals are shown in **Figure 1** (all the figures are simplified versions of some of our models, due to paper size restrictions). Goals are the basis for requirements, so the next viewpoint we developed was the Goal Refinement viewpoint, which allows to model the refinement of goals into more concrete goals, and its refinement into requirements that describe the properties that are needed to realize the goals [27]. Both of these views were based on our earlier ITIL motivation models.

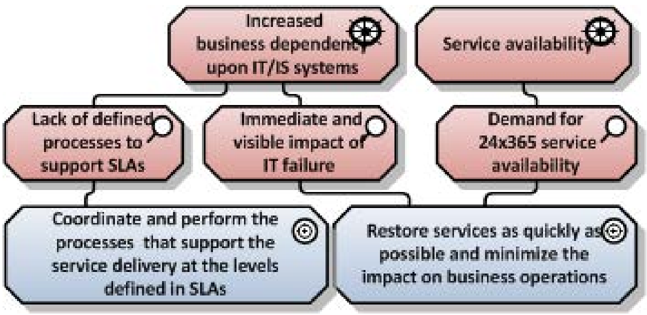


Fig. 1. Detail of Business Goals and Principles

Our next model was the Introductory View, where a simplified notation is used at the start of a design trajectory [27]. Next, we moved on to *Phase B: Target Business Architecture and Gap Analysis* where we show how the target architecture realizes the key business requirements. For this purpose, TOGAF specifies a Business Footprint diagram. In ArchiMate, this can be expressed using the Requirements Realization viewpoint, which allows the designer to model the realization of requirements by the core elements [27] (**Figure 2**).

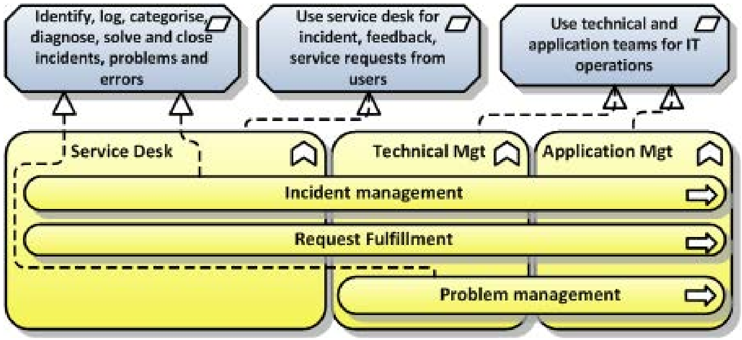


Fig. 2. Detail of Requirements Realization viewpoint

Still on this phase we also show the results of a global gap analysis for the business architecture (**Figure 3**). In both of these views we used the elements from the business layer of our core ITIL models (in light color), integrating them with ArchiSurance models (in dark color) in this latter view.

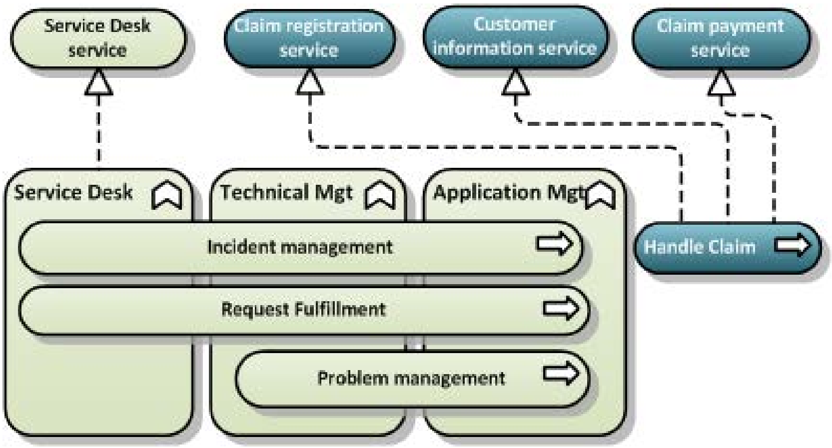


Fig. 3. Detail of target Business Architecture

To formalize the gap analysis we used Rolland's [30] gap typology. Therefore, since we are adding ITIL components to the existing EA, our gaps are mostly "Structural Changes" as they correspond to a modification of the set of elements which composes the model, and are specified by the "AddComponent" operator.

Afterwards, we moved on to *Phase C: Target Application Architecture and Gap Analysis*, where the Application Communication diagram (**Figure 4**) shows the proposed target situation for the application landscape, with the results of a global gap analysis for this layer. In the front office, shared service center, and

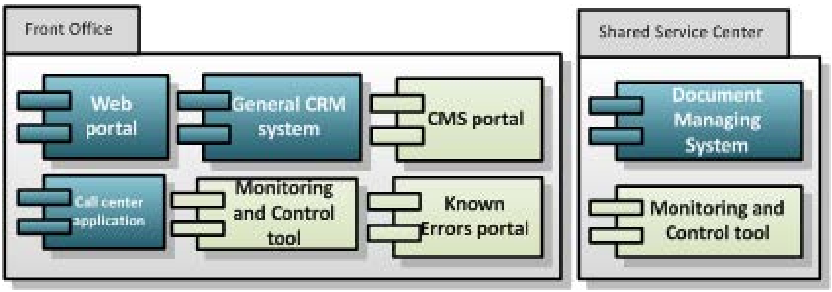


Fig. 4. Detail of target Application Architecture

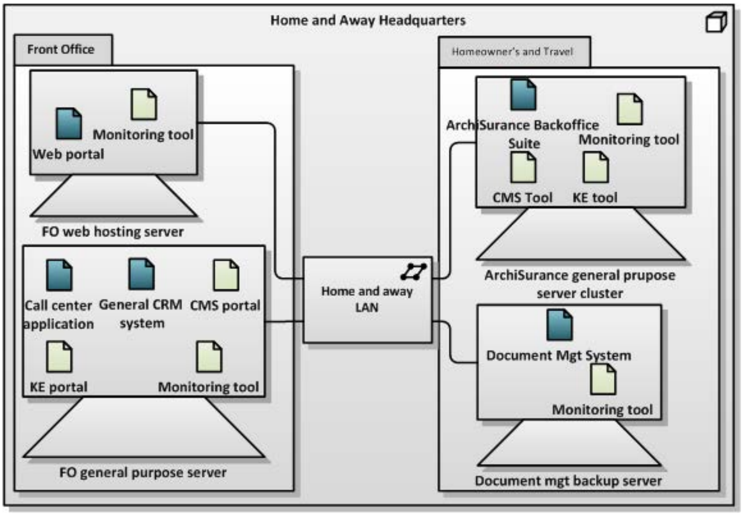


Fig. 5. Detail of target Infrastructure Architecture

back office several ITIL component applications were introduced, like the CMS portal or the Monitoring and Control Tool, with the latter being used to monitor all ArchiSurance baseline applications (in the figure we omitted the relationships for clarity sake). Next, it was time for *Phase D: Target Technology Architecture and Gap Analysis*, where we use the Infrastructure viewpoint to show the target situation for the infrastructure (**Figure 5**). Here we introduced ITIL artifacts as the CMS portal or the KE portal which are deployed on the existing (baseline) ArchiSurance infrastructure.

The following step, for Implementation and Migration Planning, TOGAF 9 introduces for Phases E and F the transition architecture, representing a possible intermediate situation (plateau) between the baseline and the target architecture. We used ArchiMate's Migration viewpoint to show the baseline, target, and transition architectures, as well as their relationships. Finally, transition architectures enable the planning of implementation projects such as Service Desk, Request Fulfillment or Problem Management. The sequence of these projects depends on which of the transition architectures is selected. This can be shown in a TOGAF Project Context which we also developed to link work packages to the functions, services, processes, applications, data, and technology that will be added, removed, or impacted by the project.

Thus, we demonstrated both of our proposals: 1) ArchiSurance is an organization with a EA representation and an ITIL implementation, where the ITIL components (and relationships) are subsets (in every layer) of the EA ones; and 2) we implemented ITIL on an organization represented by an EA, using an EA method as if it was any other architecture change.

6 Evaluation

For evaluation we will use the **The Moody and Shanks Framework** [31] which proposes the following quality factors: completeness, integrity, flexibility, understandability, correctness, simplicity, integration and implementability.

Hence, for **completeness** we can say our ArchiSurance models contain all user requirements, because these were to implement Service Operation (SO), and our ITIL models contain all the relevant SO elements and relationships. For **integrity** our models have all the SO rules and constraints, namely the ones that address which are the processes to be implemented, and their application and infrastructure dependencies. They also have **flexibility** because other ITIL processes can be picked from our overall models and added to ArchiSurance, without changing the model itself. As for **understandability** the concepts and structures used are ITIL, EA and ArchiMate ones, which are easily recognizable for people in these fields. In fact, and as a side note, when we evaluated our ITIL models through interviews, everyone quickly understood them. For **correctness** the ArchiSurance "as-is" models are provided by the ArchiMate team themselves, so we can presume correctness, as for our ITIL models, they were built by a method that mapped every ITIL concept to the correct ArchiMate one, followed by its representation according to every ArchiMate rule and convention. The integration with ArchiSurance was also performed according to ArchiMate and TOGAF conventions and rules. We can also find **simplicity** because in our ArchiSurance models we only show the relevant constructs for a coarse-grained analysis, omitting, for instance, events and business objects from our original ITIL models. Concerning **integration**, we can see from the final models that the ITIL elements and relationships fit and are consistent with the baseline architecture in every layer. And finally, for **implementability** we know the implementation is possible, because it is simply the implementation of ITIL Service Operation, something that is done quite often in organizations.

7 Conclusion

With this work, we tried to close this gap between EA and ITIL, and propose an integration through a common frame of reference, a graphical modeling language. In fact, like ArchiMate's own goal itself, our objective was also "to provide domain integration through an architecture language and visualization techniques that picture these domains and their relations, providing the architect with instruments that support and improve the architecture process" [16].

The practical outcome of our approach is that once one start thinking about implementing ITIL as an architecture change, one can use architecture tools and methods to perform it. With our ITIL models, an architect has now the tools to design ITSM organizations, hand-picking from our motivation models his organization's specific motivations and tracing them to the right ITIL processes to implement. He can model both the baseline and target architecture with our ITIL components, and then, through gap analysis on each EA layer, he will be able to see which people, information, processes, tools or infrastructure they will need to buy, keep, develop or change in order to reach the intended ITIL target.

On the other hand, this paper also wants to start a discussion about using EA for best practices in specific domains, which can be ITSM, but could as well be purchasing or logistics. In fact, by defining the motivations and architecture in each domain, one can design specific organizations and evaluate and assert their best practices' compliance.

In short, in times where cost and value generation are such important drivers, IT governance, more than ever, should turn organizations more effective and efficient. Therefore, we hope this contribution can help to better understand EA and ITIL, two worldwide standards, complementary on organizations, with distinct IT and organizational perspectives, yet so close that they have much more to gain from aligning together instead of walking apart.

References

1. Henderson, J.C., Venkatraman, N.: Strategic alignment: Leveraging information technology for transforming organizations. IBM Systems Journal 32, 472–484 (1993)
2. Gama, N., Sousa, P., Mira da Silva, M.: Integrating Enterprise Architecture and IT Service Management. In: 21st International Conference on Information Systems Development (ISD 2012) (2012)
3. Lankhorst, M., et al.: Enterprise Architecture at Work. Springer, Berlin (2009)
4. The Open Group: The Open Group Architecture Framework. Version 9 (2011)
5. Zachman, J.: A Framework for Information Systems Architecture. IBM Systems Journal 26, 276–292 (1987)
6. The Stationery Office: The Official Introduction to ITIL Service Lifecycle (2007)
7. Hanna, A., Windebank, J., Adams, S., Sowerby, J., Rance, S., Cartlidge, A.: ITIL V3 Foundation Handbook. The Stationary Office, Norwich (2008)
8. Hochstein, A., Zarnekow, R., Brenner, W.: ITIL as common practice reference model for IT service management: formal assessment and implications for practice. In: IEEE International Conference on eTechnology eCommerce, vol. 21, pp. 704–710 (2005)
9. Arraj, V.: ITIL: The Basics White Paper. The Stationary Office (2010)
10. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28, 78–105 (2004)

11. Greefhorst, D., Proper, E.: Architecture Principles: The Cornerstones of Enterprise Architecture. Springer, Berlin (2011)
12. Ross, J.W., Weill, P., Robertson, D.C. (eds.): Enterprise Architecture As Strategy: Creating a Foundation for Business Execution. Harvard Business Scholl Press, Boston (2006)
13. Pereira, C., Sousa, P.: A Method to Define an Enterprise Architecture using the Zachman Framework. In: The 19th Annual ACM Symposium on Applied Computing (SAC 2004), Nicosia, Cyprus, pp. 1366–1371. ACM (2004)
14. Van Sante T., Ermersj J.: TOGAF 9 and ITIL V3. White Paper (2009)
15. Jonkers, H., Proper, E., Turner, M.: TOGAF 9 and ArchiMate 1.0. White Paper. The Open Group (2009)
16. The Open Group, Archimate 2.0 Specification. The Open Group (2012)
17. Jan van Bon, et al.: Foundations of IT Service Management Based on ITIL v3. Van Haren Publishing (2007)
18. Braun, C., Winter, R.: Integration of IT Service Management Into Enterprise Architecture. In: ACM Symposium on Applied Computing, pp. 1215–1219. ACM, New York (2007)
19. Nabiollahi, A., Alias, R.A., Sahibuddin, S.: A Service Based Framework for Integration of ITIL V3 and Enterprise Architecture. In: 2010 International Symposium in Information Technology (ITSim), Kuala Lumpur, vol. 1, pp. 1–5 (2010)
20. Taylor, S., Lloyd, V., Rudd, C. (eds.): ITIL: Service Design. TSO, Norwich (2007b)
21. Thorn, S.: TOGAF and ITIL. Catalog number W071, p. 26. The Open Group, San Francisco (2007)
22. Correia, A., Abreu, F.B.: Integrating IT Service Management within the Enterprise Architecture. In: 4th ICSEA, Porto, Portugal, pp. 553–558 (2009)
23. Rolland, C.: Capturing System Intentionality with Maps. In: Krogstie, J., Opdahl, A.L., Brinkkemper, S. (eds.) Conceptual Modelling in Information Systems Engineering, pp. 141–158. Springer (2007)
24. Vicente, M., Gama, N., Mira da Silva, M.: Modeling ITIL Business Motivation Model in ArchiMate. In: Falcão e Cunha, J., Snene, M., Nóvoa, H. (eds.) IESS 2013. LNBIP, vol. 143, pp. 86–99. Springer, Heidelberg (2013)
25. Lankhorst M. and the ArchiMate team: ArchiMate Language Primer (2004)
26. Radhakrishnan, R.: Enterprise Architecture & IT Service Management - ITSM Frameworks and Processes and their Relationship to EA Frameworks and Processes. The Open Group (2008)
27. Jonkers, H., Band, I., Quartel, D.: ArchiSurance Case Study. The Open Group (2012)
28. Lankhorst, M., Drunen, H.: Enterprise Architecture Development and Modelling. Information Systems Journal 8, 1–16 (2007), Via Nova Architectura
29. Meertens, L.O., Iacob, M.E., Nieuwenhuis, L.J.M., van Sinderen, M.J., Jonkers, H., Quartel, D.: Mapping the business model canvas to ArchiMate. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing, pp. 1694–1701. ACM (2012)
30. Rolland, C., Salinesi, C., Etien, A.: Eliciting gaps in requirements change, Requirements Engineering, vol. 9, p. 115. ST Eliciting gaps in requirements change (2004)
31. Moody, D.L., Shanks, G.G.: Improving the Quality of Data Models: Empirical Validation of a Quality Management Framework. Information Systems 28, 619–650 (2003)

Model-Centric Strategy-IT Alignment: An Empirical Study in Progress

Constantinos Giannoulis[1], Jelena Zdravkovic[1], and Michaël Petit[2]

[1] Department of Computer and Systems Sciences
Stockholm University, Forum 100, SE-164 40 Kista, Sweden
{constantinos,jelenaz}@dsv.su.se
[2] PReCISE Research Center, Computer Science Department
University of Namur, Rue Grandgagnage 21, B-5000 Namur, Belgium
mpe@info.fundp.ac.be

Abstract. IT pervades all sectors of today's organizations. To support efficient business solutions, business-IT alignment has been long-time discussed as a solution. Given the complexity of achieving alignment, in our research we have hypothesized the importance of one partial possible solution, namely, the fit between strategy and information system requirements. To systematically investigate the influence of widely-used business strategy formulations, such as Porter's Value Chain, Kaplan & Norton's Strategy Maps, and others, we propose a model-centric approach to strategy-IT alignment where the strategy formulations are represented in the form of models, and mapped to requirements models. The objective of this paper is to present a pilot empirical investigation assessing if strategy-IT alignment is an issue of concern, and seeking to obtain insights from practitioners about relevance of our model-based view for strategy-IT alignment. The empirical information is collected through a well-prepared questionnaire-based survey.

Keywords: Business, Alignment, IT, Strategy, Requirements.

1 Introduction

Alignment between business and IT is a topic that has been widely addressed in literature. Already in 1961, organizational performance was defined to entail a coherence between at least the factors like strategy, structure, and technology, where it aimed at aligning the organization with the environment and internally organized resources to support this alignment [1]. Since then, alignment has been consistently acknowledged as an important open issue of a top-management concern [2], as well as from the IT side, for IT executives [3]. Within the literature, several dimensions and levels of the alignment have been identified [4] making the topic quite complex to be addressed holistically.

Business strategy is defined as the determination of long-term goals and courses of action using resources to achieve them, thus enabling organizations to enact it [5]. Attaining to a long-term business vision and the objectives following it makes

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 146–155, 2013.

strategy prone to a changing environment, varying from external opportunities and threats as well as internal strengths and weaknesses. A *business strategy formulation* refers to established text-based representations of business strategy within the discipline of *Strategic Management*.

Given this complexity, *our work lies on a subset of business-IT alignment focusing on the fit between strategy and information system requirements, which we refer to as Strategy-IT alignment.* From one end, strategy encapsulates a general undetailed plan of action, encompassing a certain period of time to achieve a vision, and thus, should be understood and communicated for an enterprise to define the means required for its successful execution. On the other end, information system requirements comprise the essential information needed to build the IT means to support and facilitate business operations intended at delivering offerings to customers. A core concern within Strategy-IT alignment is coordinating strategic initiatives and plans with Information Systems (IS).

Substantial work has been done on linking strategy to requirements for IS development, but typically in either an informal manner, where text-based strategy formulations are used to guide IS development from a high perspective, or by taking into consideration only basic elements of business strategy like the case is with current enterprise architectures and models.

Our proposal envisions a model-centric strategy-IT alignment based on modeling prevalent business strategy formulations such as Strategy Maps, Value Chain and others [6], which can then be mapped to requirements techniques, such as goal-oriented, or complement other strategy-IT alignment methods. Goal orientation enhances the association of desired functionality of systems to strategic goals by reinforcing the evaluation of system features towards business strategy, as well as strategy towards consumer preference.

The novelty of this approach lies in leveraging characteristics from model-driven development (MDD) such as automation and traceability, allowing for the propagation and assessment of features and/or changes towards strategy (e.g. strategic goals, targets and objectives for balance scorecards, value proposition, etc.), making changes less prone to creating problems. Traceability enhances maintainability and adaptability of systems with respect to strategy and requirements [7], making the impact of their modification traceable, which allows for impact assessment (e.g. system disruptions, conflicts, etc.).

The objective of this paper is to justify the theoretical argumentations of our Strategy-IT alignment research from an empirical perspective, namely, to validate that this alignment is an issue of concern, as well as to collect insights from practitioners about appropriateness of our model-centric approach. The main questions that are we going to address in the study are the following:

- Is strategy-IT alignment an issue of concern in practice?
- If so, what kind of solutions do practitioners consider addressing the issue?
- Do practitioners consider models as means to address this issue?

In this paper we present a work in progress concerning the design of an empirical study. A pilot study has been performed with the questionnaire and we report on the process followed, as well as present preliminary results. The following section includes the arguments motivating the need for our study. Thereafter the design of the study is presented, followed by the presentation of the pilot results. The paper concludes by presenting the intended remaining steps to complete the empirical study.

2 Related Work

Despite the acknowledged importance of the topic, very few solutions have been widely used by practitioners. Noteworthy exceptions include the *Strategic Alignment Model (SAM)* [8], and the *Business IT Alignment maturity model* [9]. During the past decade a strong empirical motivation for business-IT alignment has been put forward [4], [10]. Empirical analysis of Business-IT alignment mainly concerns the motivation for and desirability of business-IT alignment globally: (1) the relation between the alignment and business performance or (2) the relation between types of business strategies (such as conservative or innovative) and (3) the degree of alignment.

When it comes to Strategy-IT alignment, to the best of our knowledge, no such focused empirical basis exists to investigate the need for and the existence of an explicit strategy definition and the use of models for achieving the alignment. At best, the usefulness of specific theory-based proposals was shown on case studies [11, 12, 13, 14]. When looking into empirical studies published in business-IT alignment literature, Luftman's work, and annually publishing survey results with CIOs being respondents [15], is the most cited. However, his perspective is not concerned with Strategy-IT alignment particularly, and the focus is mostly on the alignment's maturity. Moreover, strategy-IT alignment is not only relevant to CIOs - Luftman's primary respondents, but also to all those affecting it, or being affected.

Possible ways to address Strategy-IT alignment include business-IT alignment approaches, whether that is alignment between business strategy and distinct enterprise models, or business strategy and Enterprise Architecture (EA). However, alignment approaches are falling short in two ways. From one side, business strategy formulations from *Strategic Management*, such as Porter's *Value Chain* [16], *Strategy Maps and Balanced Scorecards* (SMBSC) [17], and others, are overlooked in [2, 8, 11, 18]. On the other hand, EA proposals such as Zachman [19], ARIS [20], or TOGAF [21], providing the principles, methods, and models used to design and realize an enterprise's organizational structure, include business elements without linking them to business strategy formulations.

3 Overview of a Model-Centric Strategy-IT Alignment

Introducing strategy to system's design and architecture using models has a potential to facilitate the propagation of strategic notions to system development techniques and methods. In our effort to align business strategy with IT through enterprise architecture and system requirements, we have formalized established business strategy formulations as conceptual models. Conceptualizations of the strategy formulations are designed for each of the three views identified by Barney [22]:

- For *the resource-based view*, where internal capability shapes strategy, a conceptualization of Strategy Maps and Balanced Scorecards [17] exists in the means of the SMBSC meta-model [23].
- For the industrial organization, where positioning based on *competition* shapes strategy, a conceptualization of the Value Chain [16], the Value Shop and the Value Network [24] exists in the means of the Value Configuration (VC) meta-model [25].
- For the "Schumpeterian view of competition", where *innovation* shapes strategy, a conceptualization of the Blue Ocean Strategy [26] exists in the means of the BOS Meta-model [27].

Furthermore, we have designed a Unified Business Strategy Meta-Model (UBSMM), a unified conceptualization of the strategy formulations mentioned above [6], to facilitate the alignment of different business strategies, and can also serve as a pivot model between different business strategy formulations of a single or multiple enterprises.

For the scope of this study the conceptualization of SMBSC is used as an illustration of our approach [23]. A strategy map is a business strategy formulation serving as a mediator between the mission, core values, and the vision of a business to the work performed.

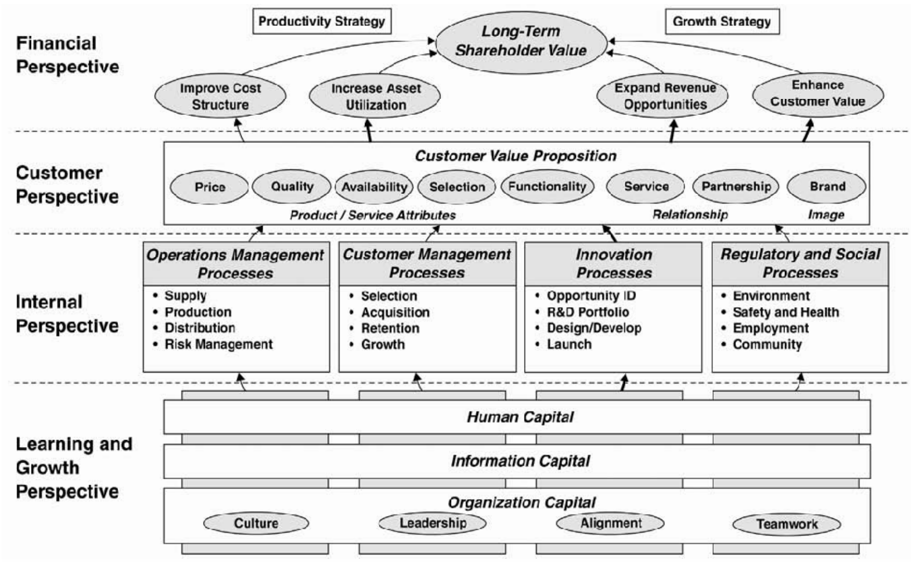


Fig. 1. The Strategy Maps and Balanced Scorecards template [17]

Kaplan and Norton proposed a template for strategy maps representing how an organization can create value [17]. A set of goals is defined and initially grouped within the financial and customer perspectives, along with goals for all types of capital, both human and economic. Goals are extended to a set of targets using measures to evaluate their achievement, and thereafter, initiatives are identified to achieve the targets, the balanced scorecard. This extension of the strategy map is the *balanced scorecard*, which is essential for monitoring and assessing the cause-effect links between strategic goals across an organization. Scorecards consist of strategic objectives and related measures, which include concrete targets and initiatives towards their achievement. Scorecards present an organization's business activities through a number of measures typically from four organizational perspectives: financial, customer, internal, learning and growth, and provides a language to communicate priorities within an enterprise. The figure below presents our current conceptualization of SMBSC, originated from [23].

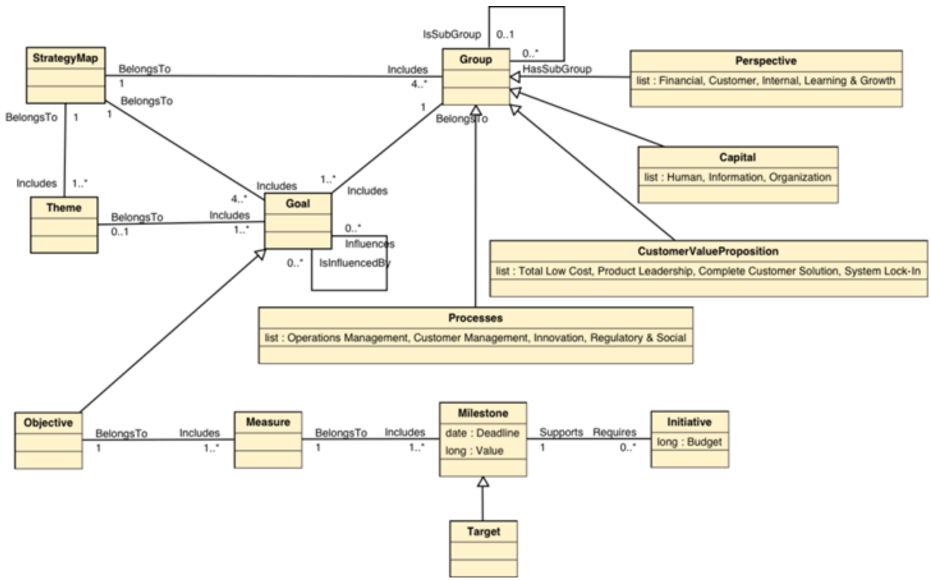


Fig. 2. The Meta-model of Strategy Maps and Balanced Scorecards [23]

Once a strategy formulation is conceptualized, it becomes possible to map it to an enterprise model, such as a goal model. For example, the mappings from the SMBSC conceptualization from Figure 2 to the i* goal modeling technique has been proposed in our previous work [28]. The i* is known for being used in the early phases of requirements engineering by focusing on the social aspects of systems by capturing the intentionality and rationale of actors within an organizational setting. It additionally supports derivation of system requirements, as presented in the OO-Method [29], where it is used to capture the organizational context and the actors' intentions for a business requirements model, as the basis for the elicitation of a conceptual model capturing the functional requirements of a system.

In addition to the theoretical proposal briefly described in this section, the objective of this paper is to present a pilot empirical investigation assessing if strategy-IT alignment is an issue of concern, and seeking to obtain insights from practitioners about relevance of our model-based view for strategy-IT alignment. The empirical information is collected through a well-prepared questionnaire-based survey.

4 Designing an Online Questionnaire for Strategy-IT Alignment

Designing, conducting and reporting an empirical study constitutes a complex operation involving several steps. In our in progress study, we have followed Öppenheim's framework [30] with fourteen stages for a social study, among which we have completed first eight, as we present in this section.

Stages one through four refer to looking into literature, reflecting upon it, and choosing an appropriate form for the study. Our study aims at validating strategy-IT alignment with an empirical basis and collecting insights about using models to address it. Due to the need for a large reach to practitioners, the design of study selected is an online questionnaire.

The assumption to be investigated (stage five) is expressed through the three questions formulated in Introduction.

For the questionnaire design (stage six) guidelines from [31], [32] have been used. These include establishing objectives, measure and scales to be used, types of questions, layout, wording, flow of questions, and identifying validity concerns. We have set the following two objectives for the questionnaire:

1. Identify whether strategy-IT alignment is a concern for the Business and IT actors within a company, and particularly for strategy and requirements for systems development (to whom, and why).
2. Identify whether an understanding gap exists between the Business and IT, why, does it affect strategy-IT alignment, and how could it be addressed.

The questionnaire was built with the free online tool *Survey Gizmo* (accessible at *http://www.surveygizmo.com*), and consists of 41 questions spreading across six sections. Sections are derived from the questions discussed in the introduction; sections one to three are related to the first two questions, focusing on the topic holistically (section one) but also on the main variables, with section two on strategy and section three on systems development. Sections four and five are related to the third question.

- The first section focuses on strategy-IT alignment capturing whether it is a concern, and whether this concern is being addressed (and how), etc.
- The second section focuses on business strategy and IT strategy.
- The third section focuses on the alignment of business strategy and systems development.
- The forth captures the respondents' background knowledge and familiarity with models and is intended to be a way to assess responses of section five.
- The fifth section focuses on the use of models for strategy-IT alignment.
- The sixth captures demographic information.

All sections include examples and explanations of core terms used, together with information motivating some questions. Types of questions used, include open-ended, multiple choice, checkboxes, and *Likert* scale questions. They cover possible alternatives making sure options offered are relevant to the questions, do not overlap, units/scales are consistently used, and double-barreled questions are avoided [32]. Questions have been neutrally formulated to avoid bias, and questions for consistency checking across answers have been used [32]. The questionnaire is available at: *http://edu.surveygizmo.com/s3/1185501/Strategy-IT-Alignment-Pilot-Study*.

The questionnaire has been tested through a pilot study (step seven) for assessing its validity and understandability. This has been done using a group of four academic experts on business-IT alignment, and a convenience sample of 52 professionals from around the world. Domain experts piloting the questionnaire have confirmed conformance to the hypothesis defined (construct validity), and sufficient domain coverage (content validity). Using a sample of professionals has simulated a real setting, which allowed testing the questionnaire by providing input on all functional aspects (language, structure, layout, etc.). The response rate was 52% (from 100 messages sent via email and through *Linkedin*). Moreover, the pilot study has offered preliminary results discussed in the next section.

Step eight entails sampling, including method and drawing. Our sampling approach includes input both from a large population across different companies, but also input from whole companies. For the former, we have selected a simple random sample of companies registered in Kista, Sweden, which includes a population of 1500 companies across multiple domains and lines of business (information provided by the *Swedish Agencies registration Office*, *http://http://www.bolagsverket.se).* In both cases companies employ at least 10 people.

5 Results from the Pilot Study

The pilot study has provided results with respect to the objectives defined. Results are preliminary as their collection purpose has been to improve the questionnaire and not to draw solid conclusions. This can happen only after the actual study will have been conducted, as it requires information that includes bias and error in respect to the target population and sample used. Therefore, validity of the results to support any conclusions drawn from the pilot study is not conclusive, but only indicative. When the study is completed, statistical analysis of the results entails taking into account several perspectives such as demographics and through descriptive statistics present data for the sample either through univariate or through bivariate analysis.

Figure 3 presents answers for the question "*Is Strategy-IT Alignment an issue of concern for your company?*" based on respondents' functional role (columns 2-4), on the job function (columns 5,6), and on the size of the company (columns 7,8).

Regarding the first objective, 56.2% confirmed that strategy-IT alignment is an issue of concern for their company (first column in Figure 1). 50% of the respondents acknowledge business utilizes the capacities of IT, but 65% thinks they do not work in sync. While 85% think that IT adds value to the business, only 25% think it contributes to strategic decision-making. Moreover, respondents are split about whether strategy insights are communicated adequately to IT (33,3% positive, 27,8% negative). Though 55.3% acknowledge IT is able to derive requirements from strategy a greater 61.1% claims that strategy can be traced to information systems.

Regarding the second objective, in terms of information dissemination between the business and IT, 54,6% think communication between the Business and IT is hindered due to the difference in information representation, 40% think strategy is understood by all. When it comes to using models for strategy-IT alignment, 50% of respondents reacted positively that it would improve communication between the Business and IT and 61.1% think it would lead to improvements of Strategy-IT Alignment. Finally, 77,7% disagreed with the statement that “strategy expressed in the form of a model would not bring any value to their company”.

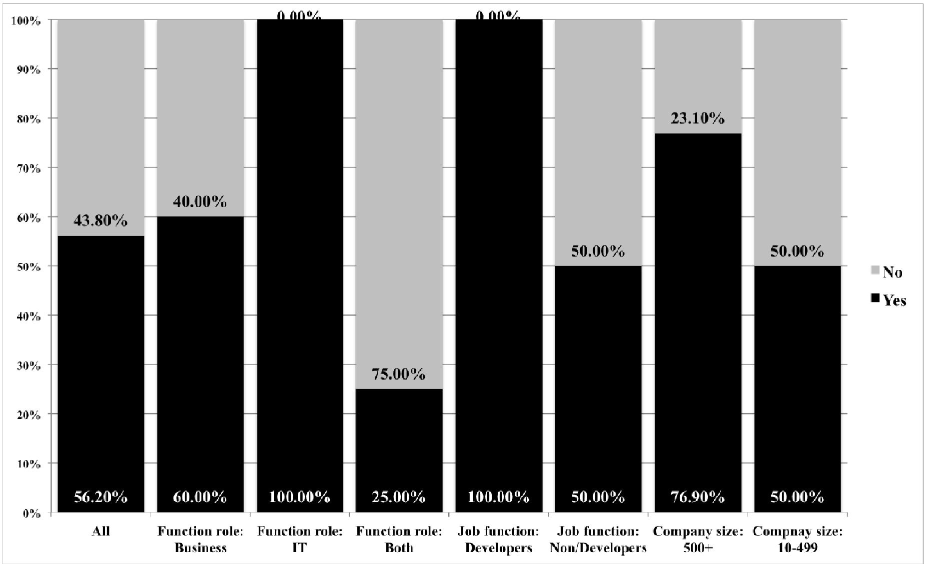


Fig. 3. Is Strategy-IT Alignment an issue of concern for your company? Answers coming from different demographic groups

Considering the research questions presented in the introduction, the results indicate strategy-IT alignment is an issue of concern for practitioners, with an interesting variation depending on respondents’ functional role, job and also size of company employed (indicated by Figure 1). However, no concrete method or technique has been identified as a solution practiced to address the issue. The sample also indicated an increased familiarity both with distinct business strategy formulations as well as with various types of models, with only a very small portion of respondents not involving models in their work. Finally, results have indicated that the use of models would lead to improvements in terms of communication between the business and IT, as well as to strategy-IT alignment. This indicates that a model-centric proposal would be well received by practitioners for strategy-IT alignment bringing value to their company.

6 Conclusion and Future Work

In this paper we have argued for the need to enrich the current body of knowledge in strategy-IT alignment with an empirical basis. We proposed the conduct of a study in the form of a questionnaire. We presented the design steps required methodologically and also reported on the current status of our work by explaining our steps and presenting sample results of the pilot study we have carried out. The study complements our theory-based research on a model-centric Strategy-IT alignment, presented in the previous work [6, 23, 25, 27, 28].

Apart from addressing the objectives of our research, the questionnaire can be also used by companies and organizations as a mechanism to assess their progress on strategy-IT alignment periodically. Feedback received from the pilot study both by domain experts as well as by the sample population involved, has provided us with suggestions for improvements important to assure the quality of the final questionnaire both in the terms of functionality and clarity but also in terms of conceptual soundness. This will constitute the core of our future work to complete the questionnaire and carry out the empirical study.

References

1. Burns, T., Stalker, G.: The Management of Innovation. Tavistock, London, UK (1961)
2. Luftman, J.N., Kempaiah, R., Nash, E.: Key issues for IT executives. MIS Quarterly Executive 5(2), 81–101 (2005)
3. Tallon, P., Kraemer, K.: Investigating the Relationship between Strategic Alignment and Business Value: The Discovery of a Paradox, pp. 1–22. Idea Group, Hershy (2003)
4. Chan, Y., Sabherwal, R., Thatcher, J.: Antecedents and outcomes of strategic IS alignment: an empirical investigation. IEEE Transaction Engineering Management 51, 27–47 (2006)
5. Chandler, A.D.: Strategy and Structure. The MIT Press, Cambridge (1962)
6. Giannoulis, C., Zdravkovic, J., Petit, M.: Model-driven strategic awareness: From a unified business strategy meta-model (UBSMM) to enterprise architecture. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Wrycza, S. (eds.) EMMSAD 2012 and BPMDS 2012. LNBIP, vol. 113, pp. 255–269. Springer, Heidelberg (2012)
7. Atkinson, Kühne: Model-Driven Development: A Metamodeling Foundation. IEEESoftware 20(5), 36–41 (2003)
8. Henderson, J.C., Venkatraman, N.: Strategic Alignment: Leveraging information technology for transforming organizations. J. IBM Systems 32(1), 4–16 (1993)
9. Luftman, J.N., Bullen, C.V., Liao, D., Nash, E., Neumann, C.: Managing the Information Technology Resource: Leadership in the Information Age. Pearson Prentice Hall (2004)
10. Aversano, L., Grasso, C., Tortorella, M.: A Characterization Framework for Evaluating Business/IT Alignment Strategies. In: Proceedings of the 14th International Conference on Enterprise Information Systems (ICEIS 2012), pp. 155–164 (2012)

11. Singh, S.N., Woo, C.: Investigating business-IT alignment through multi-disciplinary goal concepts. Requirements Engineering 14, 177–207 (2009)
12. Basili, V.R., Lindvall, M., Regardie, M., Seaman, C., Heidrich, J., Munch, J., Rombach, D., Trendowicz, A.: Linking Software Development and Business Strategy Through Measurement. Computer 43(4), 57–65 (2010)
13. Gordign, J., Akkermans, J.: Value-based requirements engineering: exploring innovative ecommerce ideas. Requirements Engineering 8, 114–135 (2003)
14. Thevenet, L.-H., Salinesi, C.: Aligning IS to organization's strategy: The INSTAL method. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007. LNCS, vol. 4495, pp. 203–217. Springer, Heidelberg (2007)
15. Luftman, J.N., Derksen, B.: Key issues for IT executives 2012: Doing More with Less. MIS Quarterly Executive 11(4), 207–218 (2012)
16. Porter, M.E.: Competitive Advantage: Creating and Sustaining Superior Performance. Free Press, New York (1985)
17. Kaplan, R.S., Norton, D.P.: Strategy Maps: Converting Intangible Assets into Tangible Outcomes. Harvard Business School Press, Boston (2004)
18. Campbell, B.: Alignment: Resolving ambiguity within bounded choices. In: 9th Pacific Asia Conference on Information Systems (PACIS 2005), pp. 1–14 (2005)
19. Zachman, J.A.: A Framework for Information Systems Architecture. IBM Systems Journal 26(3), 276–292 (1987)
20. Scheer, A.-W.: ARIS-Business Process Frameworks. Springer, Berlin (1999)
21. The Open Group: The Open Group Architectural Framework (TOGAF) Version 9. The Open Group, Reading, UK (2009), http://www.opengroup.org/togaf/
22. Barney, J.B.: Firm Resources and Sustained Competitive Advantage. Journal of Management 17(1), 99–120 (1991)
23. Giannoulis, C., Petit, M., Zdravkovic, J.: Modeling Business Strategy: A Meta-model of Strategy Maps and Balance Scorecards. In: 5th IEEE International Conference on Research Challenges in Information Science (RCIS 2011). IEEE, New York (2011)
24. Stabell, C.B., Fjeldstad, Ø.D.: Configuring Value for Competitive Advantage: On Chains, Shops, and Networks. J. Strategic Management 19(5), 413–437 (1998)
25. Giannoulis, C., Petit, M., Zdravkovic, J.: Modeling competition-driven business strategy for business IT alignment. In: Salinesi, C., Pastor, O. (eds.) CAiSE 2011 Workshops. LNBIP, vol. 83, pp. 16–28. Springer, Heidelberg (2011)
26. Chan, K.W., Mauborgne, R.: Blue Ocean Strategy. Harvard Business Review Press, Boston (2005)
27. Giannoulis, C., Zdravkovic, J.: Linking Strategic Innovation to Requirements: a look into Blue Ocean Strategy. In: Proceedings of Practice of Enterprise Modeling (PoEM 2012), CEUR 933, pp. 118–128 (2012)
28. Giannoulis, C., Zdravkovic, J.: Modeling Strategy Maps and Balanced Scorecards using i*. In: 5th International i* Workshop (iStar 2011), CEUR 766, pp. 90–95 (2011)
29. Alencar, F., Marín, B., Giachetti, G., Pastor, O., Castro, J., Pimentel, J.H.: From i* requirements models to conceptual models of a model driven development process. In: Persson, A., Stirna, J. (eds.) PoEM 2009. LNBIP, vol. 39, pp. 99–114. Springer, Heidelberg (2009)
30. Oppenheim, A.N.: Questionnaire Design, Interviewing and Attitude Measurement, Continuum, NY (1992)
31. Parasuraman, A.: Marketing Research. Addison-Wesley (1991)
32. Rattray, J., Jones, M.C.: Essential elements of questionnaire design and development. Journal of Clinical Nursing 16, 234–243 (2005)

IT Governance Mechanisms Patterns

Rafael Almeida, Rúben Pereira, and Miguel Mira da Silva

Department of Computer Science, Instituto Superior Técnico, Lisbon, Portugal
{rafael.d.almeida,rubenfspereira,mms}@ist.utl.pt

Abstract. Information Technology (IT) has been used in large organizations since the 1950s, for internal and external purposes. The pervasive use of technology in organizations has created a critical dependency on IT that calls for a specific focus on IT Governance (ITG) that is essential to ensure the Business/IT alignment objectives. However, determining the right ITG mechanisms remains a complex endeavor. Therefore, we propose to perform an exploratory research and analyze several ITG case studies to elicit possible ITG mechanisms patterns used in specific organizational context. It should be noted that a pattern is something which describes a successful solution to a problem in a specific context. Our main goal is to build some theories (ITG mechanisms patterns) which we believe that will guide organizations about the suitable ITG mechanisms to implement. The research methodology adopted was Design Science Research. We finish our research with limitations, contribution and future work.

Keywords: IT Governance, Mechanisms, Patterns, Design Science Research, Organizational Context.

1 Introduction

Information Technology (IT) has become crucial to the support, sustainability and growth of the business [1]. This pervasive use of technology has created a critical dependency on IT that calls for a specific focus on IT Governance (ITG) [2][3].

ITG define and spread the necessary mechanisms as a means of rationalizing, directing and coordinating an organization's IT-related decision making [4], in order to ensure the present and future Business/IT alignment objectives [5][6]. Some authors even state that ensuring the alignment between business and IT is one of the primary goals of ITG [7].

Recent studies have focused in some ITG problems as the inconsistencies and incongruities about the ITG mechanisms [8] or the lack of consensus about the ITG definition [9]. However, less research can be found about how organizations can effectively implement ITG [10] and the identification of the relation of ITG mechanisms with a specific organizational context is also advisable.

Therefore, this article - through the analysis of several ITG Case Studies - aims to elicit a set of ITG patterns taking into account the ITG mechanisms used by organizations as well as their organizational context. Such patterns enable the solution

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 156–161, 2013.

of "real world" problems because they capture and allow for the reuse of experiences of best practice in a specific professional domain [11].

2 Research Method

The research methodology used in this article was Design Science Research (DSR) that began growing in popularity in Information Systems (IS).

From the four artifacts produced by DSR (constructs, models, methods and instantiations) we will focus on constructs and models. Therefore, the constructs that we propose are the domain definition, the ITG mechanisms and the ITG factors. The model that we propose is the definition of ITG patterns taking into account the integration of the constructs.

As advised by [12], the research methodology applied is divided according to the two processes of DSR in IS: Build and Evaluate. In the Build process we created our constructs and model and in the Evaluate process we validated our artifacts through Hevner guidelines [13] and the appraisal of scientific community.

In order to elicit our constructs we used an extensive literature review (LR). Our main LR sources were the IEEE, ACM, and Springer digital libraries where we looked for terms as "IT Governance", "IT Governance mechanisms", "IT case study", "IT Governance factors", "Structures", "Processes" and finally "Relational" mechanisms in articles from no further than 2012.

3 Related Work

So far, few researches focused on any kind of ITG patterns elicitation. Indeed, we only found two related researches [5] and [6] and they do not solve our research problem.

Among the literature several authors argued that organizations should use ITG mechanisms [3][6][14] as advised by [15] but few researches attempt to describe and provide a complete explanation on ITG mechanisms.

We looked into several ITG mechanisms researches. The most detailed researches regarding ITG mechanisms were [6][8][14][16]. However, after a deep analysis we believe that the research [8] is the most complete one, since is grounded in an extensive literature review, try to solve some inconsistencies among the ITG mechanisms and provide a complete list of ITG mechanisms. Plus, is a recent study (2012) and all the other researches are included in their literature review references.

Therefore, we decided to adopt the list of ITG mechanisms (about 50) provided by this research. All the mechanisms are general to any kind of organization context.

Unfortunately, due to space limitations, we cannot provide the definition of the ITG mechanisms and therefore we forward the readers to the original article [8].

Determining the right ITG mechanisms is a complex endeavor and it should be recognized that what strategically works for one company does not necessarily work for another [3][17]. This means that some factors may influence the successfulness of ITG implementation.

Among the literature we found three good researches regarding ITG factors [18][19][20].

The first approach is provided by [18] and the identified factors are: Culture, Ethic, Industry, IT Strategy, Maturity, Regional Differences, Size, Structure and Trust. The second approach can be seen in [20] and the factors provided are: Overall Governance mode, Firm size, Diversification mode, Diversification breadth, Exploitation strategy for scope economies and Line IT knowledge. The third approach is provided by [19] who identified these factors: Strategic and performance goals, Organizational structure, Governance experience, Size and diversity and Industry and regional differences.

After analyzed all the different approaches, we decided to use the first approach since not only encompasses several factors presented in the other two researches detailed but is also the most recent and complete one research.

4 Results

After the identification of the ITG factors and mechanisms, we selected 30 CSs from the sources stated in Section 3.

Some of the information gathered from the 30 CSs regarding both the ITG factors and the ITG mechanisms can be seen in Table 1 and Table 2 respectively.

In Table 1 we only use "X" to indicate by which factors each organization is characterized. When all the cells regarding an ITG factor are empty, it means there was no evidence of it.

Regarding Table 2, we have adopted the following simbology: when the mechanism does not exist, the cell is empty; when the mechanism is partially implemented or there is some evidence that it is used, the cell is filled with "◖"; when the mechanism is totally implemented, we use "●".

One example about how we gathered information about ITG mechanisms is for example in CS1 "KBC documents roles and responsibilities for all hierarchical levels involved in the governance framework." we understand that they are considering the "Integration of governance /alignment tasks in roles & responsibilities" mechanism.

Unfortunately, due to space limitations, we can only provide the results of two ITG factors (Structure and Culture) and of the ITG mechanisms that are presented in the five ITG mechanisms patterns that we are going to elicit in Section 6.

Table 1. ITG Factors results

ITG Factors		Case Studies																													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Structure	Centralized		X			X	X										X			X						X	X				
	Decentralized																														
	Federal	X		X	X			X			X				X	X		X	X		X	X	X	X	X			X	X	X	X
Culture	The contest model					X		X			X						X	X	X				X	X		X	X		X	X	
	The organization as a family								X											X	X										
	The network model														X																
	The pyramidal organization									X		X	X	X																	
	The Solar system	X	X	X	X		X																								
	The well-oiled machine															X															X

Table 2. ITG Mechanisms results

		ITG Mechanisms	Case Studies																													
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Structure	1	Integration of governance /alignment tasks in roles & responsibilities	●	◖	●	●	●		●						●	●							●			●						
	2	IT strategy committee	●		●		●											●	●	●	●		●			●						
	3	IT steering Committee	●	●	●	●		●	●	●		●				●		◖	●		●	●	●	●		●	●	●	●	●	●	●
	10	IT organization structure	●	●	●	●	●	●	●			●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Processes	21	IT performance measurement	●	◖	●	●	●	●	●	●			●					●		●	●					●						●
	22	Strategic Information System Planning		●	◖	◖		●		◖	●										●	●		●					●			
	23	Frameworks ITG	●	●	●	●		◖		●	●	●	●	●	●	●				●	●											
	25	Business/IT alignment model	◖	●	●	●	●				●	●		●		●	●	●	●	●			●									
	26	Portfolio management	●	●	●	●	●	●	●	●	●	●	●			●	●	●				●	●	●	●	●	●	●	●	●		
	27	Chargeback	●		◖	●			●									●				●		●		●		●				
	28	IT budget control and reporting	●	●	●	●	●	●	●	●		●				●	●	●	●		●	●	●		●					●		
	29	Project Tracking	●	●	●	●	●		●	●	●	●	●			●	●	●		●	●	●	●	●		●	●			●		●
	31	Architectural exception process							●								◖							●		◖		●		●		
Relation	35	Shared understanding of business/IT objectives	●	◖	●	◖	◖	●								◖	◖		●	●	●		●				●			●		
	38	ITG awareness campaigns	●	●	●	●	●	◖				●				●	●		●		●	●		●				●	●			
	40	IT leadership			●							●					●		●	●	●	●				●		●	●	●		
	43	Business/IT account management	●	●	●		●															●	●		●		●	●		●	●	●

5 Discussion

Due to space limitations, we can only provide (Table 3), five of the 26 elicited ITG mechanisms patterns. These patterns were manually elicited by the authors after a deep analysis of both Table 1 and Table 2. Moreover, only when a mechanism was used in 100% of the organizations (at least 2 organizations are required) in a specific context it was considered as ITG mechanism pattern.

Table 3. ITG Patterns

	Patterns
1	Large Australian universities adopting "the contest model" culture usually use the following mechanisms: 2, 10, 21, 25, and 29.
2	Large Australian enterprises with "Federal" structure and "the contest model" culture use the following mechanism: 10, 23, 25, 29 and 40.
3	Large Belgium Insurers with "The solar system" culture use the following mechanisms: 1, 3, 10, 21, 22, 23, 25, 26, 28, 29, 35 and 38.
4	Large United Kingdom enterprises with "Federal" structure and "The contest model" culture use the following mechanisms: 10, 26, 28 and 43.
5	Large financial services American enterprises with "Federal" structure, "The contest model" culture and "IT for efficiency and IT for flexibility" strategy use the following mechanisms: 3,10, 26, 27, 29 and 31.

IT should be noted that these patterns cannot be seen as a cookbook that must be strictly followed by organizations when implementing ITG. However, they should be seen as guidance about which can be the most relevant ITG mechanisms to implement given the specific organization's context. We believe these patterns will be useful for organizations to be aware of their priorities mechanisms.

Since no other similar research was performed before, we can´t compare our results with previous scientific knowledge.

6 Evaluation

Several guidelines should be followed by researchers when they are using the DSR methodology [13]. In Table 4 we explain how our research fulfills such guidelines:

Table 3. Hevner guidelines fulfillment

	Description
1	Design as an artifact was fulfilled by producing two artifacts: A Construct and a Model. In the Construct we define the domain in which this research falls as well as the elicitation of the ITG mechanisms. The Model we constructed consists of a set of ITG mechanisms patterns regarding nine ITG factors.
2	Problem relevance was achieved by determining the relevant ITG mechanisms to be implemented according some organizational contexts as acknowledged and motivated by De Haes and Grembergen [10].
3	Design evaluation was achieved by doing a rigorous evaluation of our artifacts. This evaluation was done through the appraisal of the scientific community and by strictly following the guidelines of DSR.
4	Research contribution was achieved through the results of this research activity. These results help to improve ITG implementation successful providing ITG mechanisms patterns for specific organizational contexts.
5	Research rigor was fulfilled by the use of various methods and data collection summarized in previous sections.
6	Design as a search process is not an easy guideline to fulfill because there are no other competing approaches as we proved in Section 4.1. Likewise, as we stated, our solution is not a cookbook to be followed but a set of ITG mechanisms patterns that organizations must be aware of.
7	Communication of research was fulfilled by communicating the results of this study through the submission in reputable international conferences.

7 Conclusion

After the CSs analysis, it becomes clear that there are a set of ITG mechanisms that are comprehensively implemented by organizations. So far 26 ITG mechanisms patterns (only 5 were shown) were elicited, each one according to a specific organizational context characterized by the ITG factors. This research main contribution is the elicited ITG mechanisms patterns that can be viewed as the minimum baseline of mechanisms for each respective organizational context. Such baseline may be used by organizations to prioritize the ITG mechanisms needed for ITG implementation.

The mechanisms 3, 10, 26 and 29 are the most common among the elicited patterns and the mechanism 10 is even present in all the 26 patterns.

Of course our research has some limitations as well. The chosen factors are not static and other factors can be considered in the future as well. Plus, ethic, maturity and trust should be further detailed for more comprehensive analysis. Finally, although our difficulty in find good IT CSs among the literature, this process should be a continuous work and more CSs may be considered in the future.

For future work it will be very interesting to analyze whether the successfulness of the CSs, in order to conclude if these patterns should be reused in similar organizations. Furthermore, we intend to perform some real-world CSs in a near future to test some of our proposed theories. Finally, more interviews should also be performed in order to increase the practitioner's viewpoint.

References

1. Law, C., Ngai, E.W.: IT Business Value Research: A Critical Review and Research Agenda. International Journal of Enterprise Information Systems 1(3), 35–55 (2005)
2. De Haes, S., Van Grembergen, W.: Analysing the Relationship Between IT Governance and Business/IT Alignment Maturity. In: 41st Hawaii International Conference on System Sciences, p. 428. IEEE Press, New York (2008)
3. Van Grembergen, W., De Haes, S., Guldentops, E.: Structures, Processes and Relational Mechanisms for IT Governance. In: Van Grembergen, W. (ed.) Strategies for Information Technology Governance. IGI Publishing, Hershey (2003)
4. Huang, R., Zmud, R.W., Price, R.L.: Influencing the effectiveness of IT Governance practices through steering committees and communication policies. European Journal of Information Systems, EJIS 19, 288–302 (2010)
5. Gerrard, M.: IT Governance, a Flawed Concept: It's Time for Business Change Governance. Gartner Research (2009)
6. Weill, P., Ross, J.: IT Governance: How Top Performers Manage IT Decision Rights for Superior Results. Harvard Business School Press, Boston (2004)
7. Symons, C.: IT Governance Framework. Forrester Research (2005)
8. Almeida, R., Pereira, R., Mira da Silva, M.: IT Governance Mechanisms: A Literature Review. In: International Conference on Exploring Service Science 1.3, p. 146. IEEE Press, New York (2013)
9. Pereira, R., Mira da Silva, M.: Designing a new Integrated IT Governance and IT Management Framework Based on Both Scientific and Practitioner Viewpoint. International Journal of Enterprise Information Systems 8(4), 1–43 (2012)
10. De Haes, S., Van Grembergen, W.: An exploratory study into the design of an IT Governance minimum baseline through Delphi research. Communications of the Association for Information Systems 22(24) (2008)
11. Schadewidzt, N., Timothy J.: Comparing Inductive and Deductive Methodologies for Design Patterns Identification and Articulation. International Association of Societies of Design Research (2007)
12. March, S., Smith, G.: Design and Natural Science Research on Information Technology. Decision Support Systems 15, 251–266 (1995)
13. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quaterly 28(1), 75–105 (2004)
14. Van Grembergen, W., De Haes, S.: Enterprise Governance of Information Technology: Achieving Strategic Alignment and Value. Springer Heidelberg (2009)
15. De Haes, S., Van Grembergen, W.: IT Governance and Its Mechanisms. Information Systems Control Journal 1 (2004)
16. Peterson, R.: Information Strategies and Tactics for Information Technology Governance. In: Van Grembergen, W. (ed.) Strategies for Information Technology Governance, pp. 37–80. IGI Publishing, Hershey (2004)
17. Patel, N.V.: An emerging strategy for e-business IT Governance. In: Van Grembergen, W. (ed.) Strategies for Information Technology Governance. Idea Group Publishing, Hershey (2003)
18. Pereira, R., Mira da Silva, M.: Towards an Integrated IT Governance and IT Management Framework. In: 16th International Conference on Enterprise Distributed Object Computing, pp. 191–200. IEEE Press, New York (2012)
19. Weill, P.: Don't Just Lead, Govern: How Top-Performing Firms Govern IT. MIS Quarterly Executive 3(1), 1–17 (2004)
20. Sambamurthy, V., Zmud, R.W.: Arrangements for Information Technology Governance: a theory of multiple contingencies. MIS Quarterly 23(2), 261–290 (1999)

Correlating Services with Business Objectives in the ServAlign Framework

Aditya Ghose, Lam-Son Lê, and Evan Morrison

Decision Systems Lab, School of Computer Science and Software Engineering
University of Wollongong, New South Wales 2522, Australia
{aditya,edm92,lle}@uow.edu.au

Abstract. We present a novel approach to modeling business objectives (strategies) and a novel notion of alignment between strategy and service models leading to the successfully deployed ServAlign tool that supports automated alignment analysis.

Keywords: business objective modeling, intentional service design, strategic service alignment.

1 Introduction

The central role of the notion of *business objectives* in the theory and practice of management has long been recognized. A framework, methodology and supporting toolkit for strategic service alignment underpins services management in a variety of ways. The ability to assess alignment between the set of service offerings of an organization and its strategic objectives is critical as a "correctness check" for its operations. Alignment checking can help reveal *unrealized strategies* (i.e., strategies for which no operational support exists within the organization). It can also reveal *redundant services* (i.e., services that do not support any strategic intent). Alignment can provide the basis for service design by using the strategic landscape of an organization to guide the design of its services. The re-alignment of services to an altered strategic landscape provides the machinery for organizational response to dynamic business contexts. The alignment of strategies and business functionality has received considerable recent attention [1,2,3,4,5] but much remains to be done on 3 counts: (1) Offering a richer vocabulary for describing business objectives (2) Offering a formal definition of correlation (or alignment) between service designs and business objectives and (3) Developing automated support for the analysis of correlation or alignment. The*ServAlign* framework addresses these challenges. The tool has been implemented and evaluated successfully in industry contexts (tool and evaluation details, plus comparison with related work, ommitted here due to space constraints, can be found at www.uow.edu.au/∼aditya/servalign).

2 A Business Objective Modeling Language

We present a high-level Business Objective Modeling Language (BOML) (in the following we will use "business objective" and "strategy" interchangeably) in the

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 162–167, 2013.

following. Its expressive adequacy was validated via analysis of a large number of examples of corporate strategy documentation in the published literature, as well as about 11 actual strategy documents.

We view a *business objective model* as a set of *business objective statements* containing the following:

(1) A goal: Goals are descriptions of conditions that an organization seeks to achieve (e.g., *"Our corporate strategy is to be the market leader in mobile handsets"*).

(2) A measure (objective function or constraint): An objective function is a construct used in operations research techniques to define what the *preferred* or *optimal* solution(s) to an optimization problem might be. These are typically articulated as *maximize f* or *minimize f*, where f is a function defined on the *decision variables* (using which the constraints that *feasible solutions* are required to satisfy are also written). Our analysis of a large number of actual corporate strategy documents, as well as the management literature, suggests that strategies involving *corporate performance measures* (such as those found in the Balanced Scorecard framework) or *key performance indicators (KPIs)* are articulated in the form of maximization or minimization objectives (e.g., *"Our strategy is to maximize customer satisfaction"*). Business objectives involving performance measures are sometimes written as *constraints.* These might occur directly or may obtain from the transformation of an objective function into a constraint via the introduction of a threshold on the value of the function (e.g., one way to articulate a strategy to minimize average customer wait times at a customer care center is to agree that this strategic objective would be satisficed if the average customer wait time was 3 minutes or less). Business objectives thus articulated lead to 2 distinct ontological constructs: *(1) objective functions and (2) constraints.*

(3) A plan: A plan is a set of goals together with a set of sequencing and related coordination constraints. In the general case, a plan can be as complex as a process model (with conditional split-join constructs), but our analysis of a number of corporate strategy documents suggests that strategies are typically articulated at a very high level of abstraction, where control structures more complex than linear sequencing are almost never required.. In this paper, we will view plans only as linear sequences of goals (e.g. *"Our strategy is to first gain market acceptance in NZ, then position ourselves in the UK market, then use the UK market credibility to enter the Australian market"*). There are three steps (goals) in this strategy, connected via a linear sequencing relation.

In order to relate business objectives with service designs, we will need to refine business objectives to a point where they are expressed in a vocabulary that (approximately) matches that in which services are represented. In the following, we will refer to strategies formulated as functional goals and plans as *functional strategies.*

Objective function refinement: In an optimization problem, we seek values for a set of *decision variables*, such that these satisfy a set of *constraints* on the decision variables, and the value of an objective function is either maximized or minimized. One can view the process of computing an optimal solution as search through a space of *feasible solutions* (i.e., solutions which satisfy the set of constraints) in a manner guided by the objective function (which tells us which amongst a set of feasible solutions is most preferred). An objective function thus (implicitly) provides a *preference relation* on a set of *solutions*. In our setting, these solutions are not feasible (constraint-satisfying) assignments of values to decision variables - instead they are alternative realizations of functional strategies (i.e., mappings from functional strategies to sets of services). For now, an objective function will be *semantically* viewed (note that the *syntactic* representation remains unchanged) as a set of pairs $\{\langle S_1, S_2\rangle, \langle S_2, S_3\rangle, \ldots\}$ where each pair of the form $\langle S_1, S_2\rangle$ is a statement of preference over sets of services (in this instance, that the set S_1 is preferred to the set S_2. In the following, we will require a notion of *consistency of a set of objective functions*. Objective functions O1 and O2 are *inconsistent* if and only if there exists a pair of solutions S1 and S2 such that S1 is preferred over S2 by O1 and the reverse holds under O2 (note that this notion of consistency in contingent on the available set of solutions - O1 and O2 may be consistent given one set of solutions, but inconsistent given another). Formally, a refinement of an objective function F into a set of objective functions $\{f_1, f_2, \ldots, f_n\}$ is *valid* if and only if: (1) For any $\langle s_1, s_2\rangle \in F$, there does not exist any $\langle s_2, s_1\rangle \in f_i$ for any i. (2) $F \subseteq f_1 \cup f_2 \cup \ldots f_n$. (3) There does not exist any $F' \subset \{f_1, f_2, \ldots, f_n\}$ such that $F \subseteq \bigcup F'$. (4) $\{f_1, f_2, \ldots, f_n\}$ is consistent. The first condition in this definition ensures that the refinement of an objective function does not make statements of preference that contradict the parent objective function. The second condition states that all of the preferences statements implicit in the parent objective function must also be made by the combination of the refined objective functions (i.e., no statement of preference goes missing as a consequence of refinement). The third condition requires that the refinement is non-redundant, i.e., no objective function is included in the refinement that does not contribute to ensuring that all of the preference statements in the parent objective function are included in the refinement. Finally, we require the refined set of objective functions to be consistent.

In many situations, an objective function of the form *maximize* x can be refined to obtain $\{$*maximize* y,*maximize* $z\}$, based on the observation that $x = y+z$. Other commonly occurring patterns exist, e.g., *maximize* x refined to obtained $\{$*maximize* y, *minimize* z $\}$, based on $x = y/z$. If the variables in question are constrained to take on positive real values (as is typically the case with most QoS metrics), then refinements such as these are guaranteed to satisfy the semantics above. There are situations, however, where the explicit application of these semantics can be used to detect invalid refinements. Consider two services S_1 and S_2, in a setting where services are characterized by QoS measures m_1, m_2 and m_3, all of which we seek to maximize. Assume that m_1 is a somewhat more abstract QoS metric than the other two, such that in most (but not all) settings, the following holds: $m_1(S) = m_2(S) + m_3(S)$, for some service S. If we erroneously refine the objective function *maximize* $m_1(S)$ to obtain $\{$*maximize* $m_1(S)$,*maximize* $m_2(S)\}$, we may end up with an anomalous situation where

$m_1(S_2) < m_1(S_1)$ and $m_2(S_2) < m_2(S_1)$, but $m_3(S_1) < m_3(S_2)$. The application of the semantics above helps detects situations such as these. The ServAlign toolkit is designed to support such checking.

Following the KAOS framework, we define a refinement of a goal G into a set of sub-goals $\{g_1, g_2, \ldots, g_n\}$ to be *valid* if and only if (we assume here that a set of goals refers to the conjunction of its elements): (1) $g_1 \wedge g_2 \wedge \ldots \wedge g_n \not\models \perp$, (2) $g_1 \wedge g_2 \wedge \ldots \wedge g_n \models G$ and (3) $G' \not\models G$ for any $G' \subset \{g_1, g_2, \ldots, g_n\}$. Note that the refinement of strategies articulated as constraints is identical to the goal refinement machinery above.

We require a notion of goal (or effect) accumulation in order to define plan refinement. Plan refinement involves the refinement of the constituent goals, but with the proviso that these sub-goals are also appropriately sequenced and that the cumulative effect of achieving these goals in the specified order leads to the achievement of the parent goal. We assume without loss of generality that goals are represented in conjunctive normal form (CNF) as *prime implicates.* Given two goals g_1 and g_2 (now viewed as sets of clauses) such that g_1 is sequenced before g_2 in a plan, the cumulative goal achieved by the achievement of these two goals in sequence is given by the pairwise goal accumulation function $acc(\langle g_1, g_2 \rangle) = g_2 \cup \{g_1' \mid g_1' \subseteq g_1, g_1' \cup g_2$ is satisfiable and for any g_1'' where $g_1' \subset g_1'' \subseteq g_1, g_1'' \cup g_2$ is unsatisfiable$\}$. We use this to define a goal accumulation function over a sequence of goals, $Accm(\langle g_1, g_2, \ldots, g_n \rangle)$. If $Plan = \langle g_1, g_2 \rangle$, then $Accm(Plan) = acc(Plan)$. If $Plan = \langle g_1, g_2, \ldots, g_n \rangle$ is a sequence with more than two goals, then $Accm(Plan) = acc(Accm(\langle g_1, g_2, \ldots, g_{n-1} \rangle), g_n)$. Note, acc and hence $Accm$ are sets of sets of clauses in general (accumulation of goals may lead to multiple mutually-exclusive non-deterministic outcomes). The refinement of a plan $P = \langle g_1, g_2, \ldots, g_n \rangle$ to a sequence of sub-plans $\langle \langle g_{11}, g_{12}, \ldots, g_{1p} \rangle, \langle g_{21}, g_{22}, \ldots, g_{2q} \rangle, \ldots, \langle g_{n1}, g_{n2}, \ldots, g_{nr} \rangle \rangle$ is *valid* if and only if: (1)For each $g_i \in P$, $\{g_{i1}, g_{i2}, \ldots, g_{is}\}$ is a valid (goal) refinement of g_i. (2) For every $e \in Accm(P)$, there exists an $e' \in Accm(\langle g_{11}, g_{12}, \ldots, g_{1p}, g_{21}, g_{22}, \ldots, g_{2q}, \ldots, g_{n1}, g_{n2}, \ldots, g_{nr} \rangle$ s.t. $e' \models e$ These semantic constraints on *valid strategy refinements* can be leveraged in three ways in the ServAlign toolkit. First, it provides a means of generating strategy refinements in an automated fashion that is correct with respect to these semantics. Second, it provides a mechanism for checking user-generated refinements. Finally, it provides the basis for generating a library of *strategy refinement patterns* that a user may potentially use.

3 Strategic Service Alignment: Foundations

A *strategic service architecture* consists of:
(1) A strategy model: A 4-tuple $\langle Goals, Plans, Objectives, Constraints \rangle$. We will assume that strategies at varying levels of abstraction are related via a set of strategy refinement links.

(2) A services model: This consists of a set of business services (we require that these be represented, at a minimum, in terms of formal *post-conditions* and *QoS constraints*). We do not consider the sub-structure of services in this paper, or control flow relations between them (these are addressed elsewhere). *(2) A set of strategy-service realization links:* These relate strategies to the services that realize them and can be of two kinds: *basic realization links* (which directly associate strategies, at the most refined level, with services) and *derived realization links* (such a link would associate a strategy to a service via a refinement of the former, which be directly related to the latter via a basic realization link).

We will refer to any strategy associated with a service via a realization link as a *strategic antecedent* of the service. Let Con_{Serv} be the set of QoS constraints described in any service contained in a set of services $Serv$ (we assume a uniform *global* vocabulary for QoS variables). A set of services $Serv$ is *more preferred* than another set $Serv'$ under a set of consistent objective functions O if the value of the optimal solution of the optimization problem $\langle f, Con_{Serv} \rangle$ is greater than the value of the optimal solution of the optimization problem $\langle f, Con_{Serv'} \rangle$, for every distinct objective function f in O (assuming, without loss of generality, that every objective function is a maximization objective). Given a consistent set of objective functions O, and given a functional strategy Str, an *optimal realization of Str relative to O* is any subset $Serv$ of the set of available services such that $Serv$ realizes Str and for any alternative subset $Serv'$ of the set of available services that might also realize Str, $Serv'$ is less preferred to $Serv$ under O. Basic realization links are established using the following machinery: (1) A set of services $\{s_1, s_2, \ldots, s_n\}$ *realizes* a goal g if and only if it is a minimal (with respect to set inclusion) set of services such that $Postcond(s_1) \wedge Postcond(s_2) \wedge \ldots \wedge Postcond(s_n) \models g$. (2) A set of services realizes a plan if and only if it is the minimal set of services (i.e. there is no subset that achieves the same condition) that realizes each of its constituent goals. (3) A set of services $Serv$ realizes an objective function f if and only if, for each strategic antecedent Str of a service in $Serv$ that is not an objective function or constraint, the optimal realization of Str relative to f is included in $Serv$. (4) We will use a constraint negation operator (using the standard negation symbol $\neg$) which asserts the negation of the associated constraint predicate. Thus, $\neg(x < 20)$ is $x \geq 20$, and so on. A set of services $Serv$ realizes a constraint c iff $Con_{Serv} \cup \neg c$ is unsatisfiable. Testing the satisfiability of a set of constraints thus obtained can be done using standard constraint solving techniques.

We will require a notion of *internal alignment* of a strategy model. A strategy model $\langle Goals, Plans, Objectives, Constraints \rangle$ (a 4-tuple consisting of the sets of goals, plans, objective functions and constraints respectively) is aligned iff: (1) $Goals \cup \{Accumulate(p) \mid p \in Plans\}$ is satisfiable, and (2)$Constraints$ is satisfiable and (3)$Objectives$ is consistent. We note that satisfiability checking for $Constraints$ will require the use of constraint solvers (simple solvers for linear inequalities on reals will suffice for most cases, but more sophisticated solvers may be required depending on the expressivity/complexity of the language for defining constraints on QoS factors). We note that a similar model of internal alignment for the services model is not of interest. This is because there is no requirement for concurrent execution of all of the services in a services model (unlike the strategy model - where there is an implicit assumption that an organization seeks to *concurrently* achieve *all* of its strategies). It is reasonable, therefore, for an organization to support a service which increases exposure to a certain market and concurrently support another that reduces exposure to that market - the crucial point is that these services are not executed concurrently.

Non-obstruction is another critical attribute of alignment. A service S *obstructs* a functional strategy Str under the following conditions: (1)If Str is a goal, then $postcond(S) \cup Str \models \perp$ or (2)if Str is a plan, then $postcond(S) \cup Accumulate(Str) \models \perp$.

A strategic service architecture is *aligned* iff: (1) The strategy model is *internally aligned*. (2) Every user-specified strategy (as distinct from a strategy obtained via refinement) is *realized* by some subset of the services model. (3) Every service in the services model has a user-specified *functional strategy* as a *strategic antecedent* (note that it would not make sense to consider services that exist only to realize objective functions, and no functional strategy). (4) No service in the services model *obstructs* any functional strategy in the strategy model.

Assessing strategic alignment involves progressive refinement of the user-specified strategies until these use the same vocabulary as the services model. At this point, the test for alignment of a strategic service architecture is applied. The critical question is: how do we decide when to stop refining strategies any further? Ideally, when the concept vocabulary used for the most refined set of strategies is included in the concept vocabulary used to define the services model. The problem with this approach is that if the models were inherently mis-aligned, we would continue strategy refinement without termination and never find the required *ontological match*. Our approach, instead, is to leverage the degree of ontological match. We iterate the strategy refinement step as long as there is monotonic improvement in the degree of match between the resultant vocabularies, and stop prior to the degree of match decreasing.

References

1. Halleux, P., Mathieu, L., Andersson, B.: A method to support the alignment of business models and goal models. In: Workshop on Business/IT-Alignment and Interoperability (BUSITAL 2008), Montpellier, France (June 2008)
2. Barone, D., Jiang, L., Amyot, D., Mylopoulos, J.: Composite indicators for business intelligence. In: Jeusfeld, M., Delcambre, L., Ling, T.-W. (eds.) ER 2011. LNCS, vol. 6998, pp. 448–458. Springer, Heidelberg (2011)
3. Etien, A., Rolland, C.: A Process for Generating Fitness Measures. In: Pastor, Ó., Falcão e Cunha, J. (eds.) CAiSE 2005. LNCS, vol. 3520, pp. 277–292. Springer, Heidelberg (2005)
4. Thevenet, L.-H., Salinesi, C.: Aligning IS to organization's strategy: The INSTAL method. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007. LNCS, vol. 4495, pp. 203–217. Springer, Heidelberg (2007)
5. Pijpers, V., Gordijn, J., Akkermans, H.: e^3 alignment: Exploring inter-organizational alignment in networked value constellations. Journal of Computer Science & Applications 6(5), 59–88 (2009)

Cognitive Aspects of Structured Process Modeling

(Position Paper)

Jan Claes, Frederik Gailly, and Geert Poels

Department of Management Information Science and Operations Management,
Ghent University, Tweekerkenstraat 2, 9000 Gent, Belgium
{jan.claes,frederik.gailly,geert.poels}@ugent.be

Abstract. After visualizing data of various observational experiments on the way in which modelers construct process models, a promising process modeling style (i.e., structured process modeling) was discovered that is expected to cause process model quality to increase. A modeler constructs process models in a structured way if she/he is working on a limited amount of parts of the model simultaneously. This paper describes two cognitive theories that can explain this causal relation. Cognitive Load Theory (CLT) suggests that the amount of errors increases when the limited capacity of our working memory is overloaded. Cognitive Fit Theory (CFT) states that performance is improved when task material representation matches with the task to be executed. Three hypotheses are formulated and the experimental set-up to evaluate these hypotheses is described.

Keywords: business process modeling, process of process modeling, structured process modeling.

1 Introduction

Between 2009 and 2012 several observational experiments were performed to study different characteristics of how subjects create process models (e.g., [1–9]). The focus was on relating properties of the *process of process modeling* to specific characteristics of, for example, a case description [6], the modeler [7], and the modeling result (i.e., the constructed process model) [4]. Therefore, every activity on the modeling canvas was recorded (e.g., create_activity, create_edge, move_activity, etc.) [1]. We were granted access to the data of these experiments to be able to study them in detail with the use of process mining techniques.

We developed a way to visualize in a more accessible way the raw, uninterpreted data of single process modeling instances: The PPMChart [2] displays the recorded data of the modeling process for one modeling session (see Fig. 1). Each process model element that existed during the modeling process is represented by a horizontal time line. These lines are vertically sorted according to the order a liquid would reach the elements of the process model if it is converted into a flow network. On every line, the operations on the element it represents are displayed as colored dots that are

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 168–173, 2013.

placed according to the time of their occurrence. The color of the dot represents the operation on the element (e.g., create_activity in bright green, create_edge in blue-green, create_gateway in dark green, etc.)

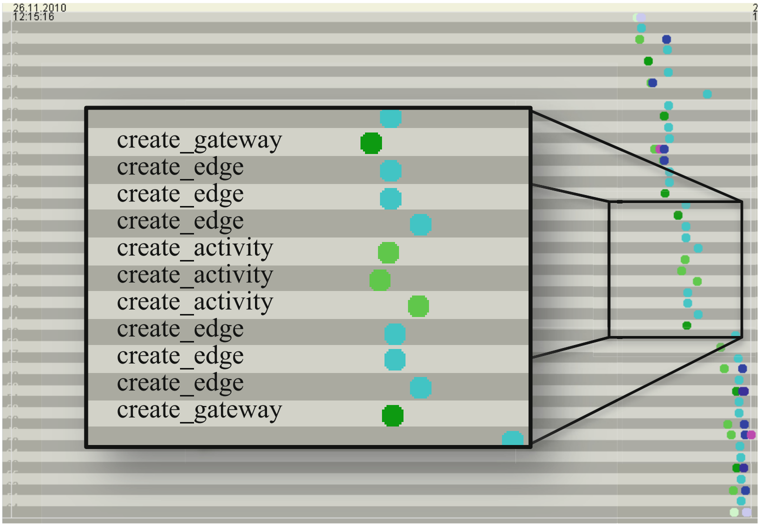


Fig. 1. Example of a PPMChart

Next, to be able to graphically discover reoccurring patterns, various PPMCharts were compared [2]. Each chart was supplemented with the quality measurement of the resulting process model (i.e., mainly soundness [10]). Three conjectures were formulated about the relation of specific properties of the modeling process to the quality of the resulting process model [4]. One of these conjectures involves the concept of structured process modeling. We call a modeling session highly structured if the modeler is working on a limited amount of process model blocks at the same time [4]. A process model block is defined as the collection of elements in two or more optional or parallel paths in the process model: i.e., a split construct, a matching join construct and all according nodes and edges of different paths between both routing constructs (see for example highlighted part of Fig. 1) [4]. The more process model blocks that are simultaneously under construction, the lower the degree of structuredness of the modeling process.

In every dataset we studied, we discovered a highly significant non trivial relation between the degree of structured process modeling and the quality of the resulting process model [4]. Therefore, we will try (i) to explain why a structured process modeling style could lead to high process model quality, and (ii) to prove that a structured process modeling style causes higher process model quality. We refer to [4] for an overview of the metrics we use for the degree of structured process modeling and process model quality.

2 Theory

In order to be able to explain why a structured process modeling style could lead to a higher quality of the resulting process model, in this section cognitive aspects of the processing capabilities of the human brain are addressed.

According to [11] people have three types of memory: sensory memory, working memory, and long-term memory (see Fig. 2). People's observations are firstly stored in the sensory memory for a very short period after which a selection of the information is automatically and unconsciously redirected to the working memory [12]. Next, the information is complemented with existing information from the long-term memory which results in the storage of newly formed long-term information (i.e., transfer) and/or directly leads to specific performances (i.e., reflexes) [13]. For the construction of a process model, the information about observations concerning the process to be modeled is recollected in the working memory (i.e., organization) and combined with other useful knowledge from the long-term memory (i.e., retrieval). Examples of useful knowledge in the long-term memory are domain knowledge, knowledge about modeling, about the modeling language, etc.

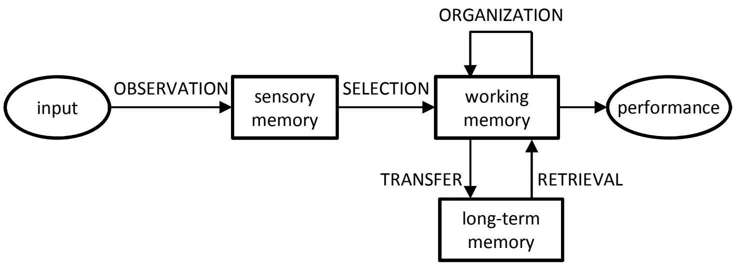


Fig. 2. Sensory memory, working memory and long-term memory

The working memory has a limited capacity of seven plus or minus two 'chunks' of information [14]. The amount of information stored in one chunk depends on the expertise of the subject on the specific task. Our long-term memory is structured in schemas that consist of connected pieces of information. People considered experts at a certain task, generally store more information (larger schemas) and have more efficient access to this information (well-structured and well-connected schemas) [15]. Therefore, they can store more information in a single chunk of working memory (one schema of an expert provides access to more information than a novice's schema). Thus, everybody's working memory has about the same capacity, but differences are related to how efficiently people use the limited working memory.

The efficiency with which our working memory is used is determined by the cognitive load of performing a task [15]. Three types of cognitive load exist: (i) intrinsic cognitive load (i.e., the amount of information needed to perform the task, depending on the task), (ii) extraneous cognitive load (i.e., the amount of information needed to interpret the input, depending on the representation of the task data), and

(iii) germane cognitive load (i.e., the remaining amount of information the subject needs to load in the working memory for performing the task, mainly depending on the expertise of the subject) [13]. CLT suggests that when people encounter a shortage of working memory they tend to make mistakes [15].

For a particular task and a particular task material representation, only the germane cognitive load influences people's performance in the execution of the task. Moreover, as discussed before, experts occupy less working memory than novices to cover this germane cognitive load. However, it is not practical to only focus on the expertise of a subject in task performance to increase her/his effectiveness and efficiency, because training a novice to become an expert takes time. Hence, a technique that reduces the necessary amount of information to be stored *at the same time* in the working memory, is an appropriate candidate to improve a human's effectiveness and efficiency to perform a certain task.

Structured process modeling is a technique that encourages modelers to work on few elements of the process model *at the same time*. Therefore, we argue that less working memory capacity is needed to model in this way than when working on several parts of the model simultaneously. This explains why structured process modeling can cause process model quality increase. Requiring less working memory capacity reduces the chance of making errors [15] and leaves more space for other activities (e.g., lay-out), which, in turn, causes quality improvement [16].

There is another cognitive theory that influences the result of a modeling task. The Cognitive Fit Theory states that when the task material *representation* fits with the *task* to be executed, people tend to be more effective and more efficient in executing the task [17]. For example, a table representation of data is argued to fit better for solving questions that ask about facts, and a graph representation fits better for questions about insightful information derived from the data [18].

Previous research indicates that a breadth-first ordering, according to the process model to be constructed, of the descriptions of activities is related to a higher model correctness than a depth-first or random ordering [6]. Note that the structured process modeling technique is similar to a breadth-first modeling approach (i.e., first finish parallel paths (breadth) before working on the consecutive parts (depth). Therefore, we suspect that a breadth-first ordering of the task description, in combination with a modeling style akin to breadth-first modeling (i.e., structured process modeling), provides the benefits of cognitive fit, and would thus consolidate the effect of structured process modeling on process model quality.

3 Hypotheses

The discussed observations (in Section 1) and theories (in Section 2) inspired us to formulate next three hypotheses:

- H1: Structured process modeling relates to process model quality improvement.
- H2: The quality improvement will be higher for novices than for experts (expertise in the case domain, in modeling, and/or in the modeling language).
- H3: The quality improvement will be higher if the task representation fits with the technique (i.e., a breadth-first ordering of the task description).

4 Research Method

In order to be able to corroborate the hypotheses, we plan to perform a double-blind, randomized, controlled experiment (see Fig. 3). The targeted subjects are a group of 150 master students that take a Business Process Management course. The students will be randomly appointed to a treatment and a control group. The treatment group (T) will be instructed to model using the structured process modeling style. The control group (C) will have a fake treatment (half of them get no instructions, half of them learn a technique that can be considered as depth-first modeling). In each group (T and C) an equal amount of participants will receive a breadth-first sorted, a depth-first sorted, and a randomly sorted task description.

The session will start with a short tool tutorial and a pre-test case to determine initial degree of structured process modeling (ST1, SC1) and process model quality (QT1, QC1). Next, the instructions of the treatment and fake treatment will be given. Finally, the experiment case has to be solved by the participants and the altered degree of structured process modeling (ST2, SC2) and process model quality (QT2, QC2) will be measured. We can check if the treatment had effect through the comparison of the degree of structured process modeling before and after treatment.

The hypotheses can be evaluated using the process model quality measurements in different subgroups before and after the treatment. To investigate H1, the results of T and C should be compared (75 students each). H2 can only be studied if there are enough domain experts for the particular cases among the students or by comparing to results from new experiments with more experienced subjects. For H3, the students from T with a breadth-first ordered text (25 students) can be compared to the other students in T (50 students), and with the part from C that has received a depth-first ordered text and the depth-first modeling instructions (13 students).

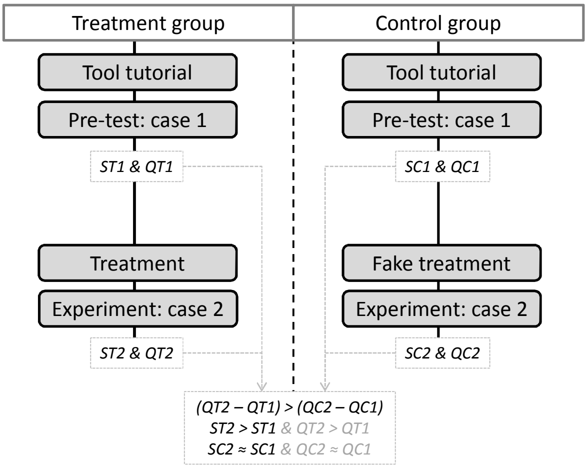


Fig. 3. Experiment set-up (S: structuredness, Q: quality, T: treatment group, C: control group) Fake treatment: depth-first modeling in one session (Ca), extra exercise in other session (Cb)

References

1. Pinggera, J., Zugal, S., Weber, B.: Investigating the process of process modeling with cheetah experimental platform. In: Mutschler, B., Recker, J., Wieringa, R., Ralyte, J., Plebani, P. (eds.) Proc. ER-POIS 2010, vol. 603, pp. 13–18. CEUR-WS (2010)
2. Claes, J., Vanderfeesten, I., Pinggera, J., Reijers, H.A., Weber, B., Poels, G.: Visualizing the process of process modeling with pPMCharts. In: La Rosa, M., Soffer, P. (eds.) BPM 2012 Workshops. LNBIP, vol. 132, pp. 744–755. Springer, Heidelberg (2013)
3. Pinggera, J., Soffer, P., Zugal, S., Weber, B., Weidlich, M., Fahland, D., Reijers, H.A., Mendling, J.: Modeling styles in business process modeling. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Wrycza, S. (eds.) EMMSAD 2012 and BPMDS 2012. LNBIP, vol. 113, pp. 151–166. Springer, Heidelberg (2012)
4. Claes, J., Vanderfeesten, I., Reijers, H.A., Pinggera, J., Weidlich, M., Zugal, S., Fahland, D., Weber, B., Mendling, J., Poels, G.: Tying process model quality to the modeling process: The impact of structuring, movement, and speed. In: Barros, A., Gal, A., Kindler, E. (eds.) BPM 2012. LNCS, vol. 7481, pp. 33–48. Springer, Heidelberg (2012)
5. Zugal, S., Pinggera, J., Weber, B.: The impact of testcases on the maintainability of declarative process models. In: Halpin, T., Nurcan, S., Krogstie, J., Soffer, P., Proper, E., Schmidt, R., Bider, I. (eds.) BPMDS 2011 and EMMSAD 2011. LNBIP, vol. 81, pp. 163–177. Springer, Heidelberg (2011)
6. Pinggera, J., Zugal, S., Weber, B., Fahland, D., Weidlich, M., Mendling, J., Reijers, H.A.: How the structuring of domain knowledge helps casual process modelers. In: Parsons, J., Saeki, M., Shoval, P., Woo, C., Wand, Y. (eds.) ER 2010. LNCS, vol. 6412, pp. 445–451. Springer, Heidelberg (2010)
7. Pinggera, J., Furtner, M., Martini, M., Sachse, P., Reiter, K., Zugal, S., Weber, B.: Investigating the process of process modeling with eye movement analysis. In: La Rosa, M., Soffer, P. (eds.) BPM 2012 Workshops. LNBIP, vol. 132, pp. 438–450. Springer, Heidelberg (2013)
8. Soffer, P., Kaner, M., Wand, Y.: Towards understanding the process of process modeling: Theoretical and empirical considerations. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM 2011 Workshops, Part I. LNBIP, vol. 99, pp. 357–369. Springer, Heidelberg (2012)
9. Indulska, M., Recker, J., Rosemann, M., Green, P.: Business process modeling: Current issues and future challenges. In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) CAiSE 2009. LNCS, vol. 5565, pp. 501–514. Springer, Heidelberg (2009)
10. Van der Aalst, W.M.P.: Verification of workflow nets. In: Azéma, P., Balbo, G. (eds.) ICATPN 1997. LNCS, vol. 1248, pp. 407–426. Springer, Heidelberg (1997)
11. Atkinson, R., Shiffrin, R.: Human memory: A proposed system and its control processes. Psychology of Learning and Motivation 2, 89–195 (1968)
12. Sperling, G.: A model for visual memory tasks. Human Factors 5, 19–31 (1963)
13. Sweller, J.: Cognitive architecture and instructional design. Educational Psychology Review 10, 251–296 (1998)
14. Miller, G.: The magical number seven, plus or minus two: Some limits on our capacity for processing information. Psychological Review 63, 81–97 (1956)
15. Sweller, J.: Cognitive load during problem solving: Effects on learning. Cognitive Science 12, 257–285 (1988)
16. Schrepfer, M., Wolf, J., Mendling, J., Reijers, H.A.: The impact of secondary notation on process model understanding. In: Persson, A., Stirna, J. (eds.) PoEM 2009. LNBIP, vol. 39, pp. 161–175. Springer, Heidelberg (2009)
17. Vessey, I., Galletta, D.: Cognitive fit: An empirical study of information acquisition. Information Systems Research 2, 63–84 (1991)
18. Vessey, I.: Cognitive fit: a theory-based analysis of the graphs versus tables literature. Decision Sciences 22, 219–240 (1991)

Zooming In and Out in Requirements Engineering

Manuel Imaz

BlendMind Madrid, Spain
imaz@mac.com

Abstract. In this paper we present some cognitive guidelines to move from higher degrees of abstraction to lower, more concrete degrees of abstraction, and reciprocally. For a time, in Requirements Engineering this approach was called the *top-down method* and it was intended to be used as a general method in the first steps of specifying a system: the whole application was decomposed using this method. In everyday analysis and design, it continues to be a fundamental way of separation of concerns, applying the motto of *divide and conquer*. In order to divide, we must visualize the whole system in terms of lower level components: processes, use cases or user stories. But the important question is that zooming in can not start –in Requirements Engineering– at every level but at a given level of abstraction, usually at the level where some event will trigger the execution of some processes: such is the case of user stories and use cases. Another important issue we want to stress is the essential role that stories play in our cognition: this is why user stories –as well as use cases– have been and continue to be intensively used in software development.

Keywords: zooming in, zooming out, decomposition, abstraction level, top-down.

1 Introduction

The traditional structured analysis and design methods which appeared in the last '70 promoted –for requirements engineering– the top-down method, which allowed to move from higher levels of abstraction to intermediate ones and finally to lower levels of abstraction. This idea, inherited from the structured programming field, had the underlying conception that moving between abstraction layers was something that we could arbitrary do without any limitation.

Unfortunately for the top-down method, some authors [22] detected it was, in fact, a *myth*. In fact, the *method* can be applied by analysts when they have already gained experience with previous similar systems they have already analyzed and designed (at least in the early steps), so they are, in fact, applying their previous knowledge and are not discovering the internal structure of new systems by going top down. Analysts had, this way, the delusion of applying a real method instead of observing that they were working using their skills

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 174–185, 2013.

and retrospectively adjudicating the *success* to the supposedly well established top-down method.

In Imaz [10] we stated that:

> *Starting at a high level, we conceive a whole large system implying composition by other subsystems. We are a long way from basic-level categories. It is at this level where there is a risk of loosing one's way, as Ed Yourdon ([22], p. 360) has pointed out in relation to the top-down problem: "analysis paralysis," the "six analyst" phenomenon, or the "arbitrary physical partitioning". Yourdon's proposal is to use a middle-out method, in which the middle is closer to basic-level categories, and from where to go down and up looking for concepts that fit the analysis purpose better (see fig. 1). Moreover, somebody who successfully applies the top-down method is a person who has sound skills in the specific domain that one is working on.* [10] p. 86

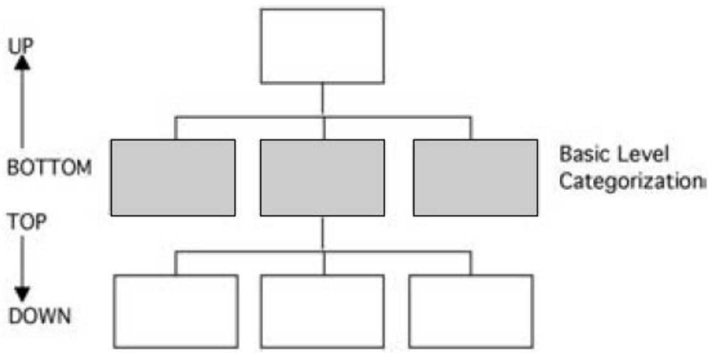


Fig. 1. Two different movements: bottom-up and top-down

What Yourdon pointed out about top-down method was in line with the categorization theory that Eleanor Rosch established in the mid-1970s, for which she tried to offer an empirical foundation. One important result of Rosch's [12] [16] work is the concept of basic-level categories. Basic-level categories are at a middle level of abstraction. They refer to subcategories or category members that have a special cognitive status: that of being a best example.

Basic-level categories are basic in a number of respects. From a perceptual point of view, basic-level categories map to an overall shape, to single images with fast identification. From a functional point of view, basic-level categories are associated with a general motor program –the way in which our brains control our movements– for using, for example, chairs and tables. In terms of mental images, we can form a general mental image of basic-level categories like chairs or dogs. But we cannot form a mental image for superordinate categories like furniture or animals.

Matching the points raised by Yourdon (what we can categorize are processes at an intermediate level, which are launched by some concrete events), in the middle '90 was coined the concept of *use case* and with the Agile movement appeared a similar construct: that of *user stories*. Both constructs appear as solid basic-level categories –at an intermediate level– and their success until now provide the empirical evidence that they are solid requirement engineering constructs due to this intermediate level conceptualization. In the following sections we will give some additional cognitive arguments and foundations.

2 Cognitive Semantics

In this section we will show how the concept of use case is based on some cognitive semantics' ideas of perspective, viewpoint, focusing and so on.

Leonard Talmy states that Cognitive Semantics is the study of the way conceptual content is organized in language. In Talmy's view, a sentence (or other portion of discourse) does not objectively represent its referent scene –the represented scene is not something out there in the world–, but it evokes in the listener a cognitive representation, defined as an emergent, compounded by various cognitive processes out of the referential meanings of the sentence elements, understanding of the present situation, general knowledge, and so on [19] p. 93, note 2.

Historically, science has tried to show its objectivity eliminating the subject from the scientific discourse. The same effort has been assumed by the software engineering community when using a disembodied discourse, but this intention is unmasked when analyzing in detail some conceptual structures in which the subject surreptitiously reappears, as –for example– the concept of *perspective* implies an object and a subject and the concept of *focusing* implies that the subject is using his visual capacity (as Langacker defines it: "what we choose to look at" [13]).

Another aspect of cognitive semantics is that the conceptual structure is *embodied*, that is, the nature of the human mind is largely determined by the form of the human body. But the *form* of the human body must be understood in a broad sense, meaning the human being in an environment, in a given situation –cultural, social, and so on– as some concepts of CS imply.

In terms of literary analysis it is said that a narration has a content but, equally important, it has also a style. Analyzing the question from the point of view of Cognitive Semantics (in particular, Langacker's Cognitive Grammar), it can be stated that:

> *In viewing a scene, what we actually see depends on how closely we examine it, what we choose to look at, which elements we pay most attention to, and where we view it from. The corresponding labels I will use, for broad classes of construal phenomena, are* ***specificity****,* ***focusing****,* ***prominence****, and* ***perspective****. They apply to conceptions in any domain.* [13] p. 66 (bolds in the original)

Associated with the concept of scene, for example, we have the concepts of focusing, perspective and so on. It is evident that a perspective implies a subject observing a scene from a given point of view, that is, a subject in a given situation.

The concept of *perspective* allows us to make a difference between observing a computer system in the organization or observing the enterprise itself, and conceptualizing the internals of the computer system or the business processes running into the enterprise. The concept of perspective is represented in (Fig. 2):

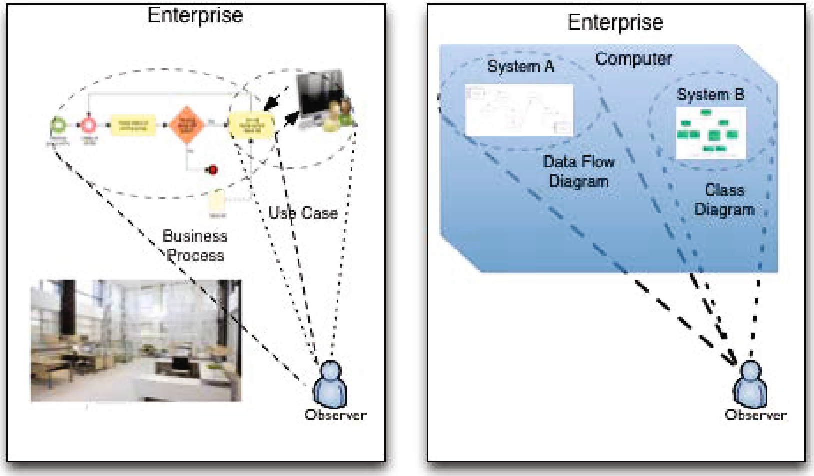


Fig. 2. Two different scenes: the enterprise and two systems into a computer

When focusing on the computer system we need additional concepts in order to use different categories applied to the same system. Besides what we choose to look at –focusing– we need to take into consideration which elements we pay most attention to or *prominence*, in particular one sort of prominence: *profiling*. Langacker states that:

> *As the basis for its meaning, an expression selects a certain body of conceptual content. Let us call this its conceptual* ***base****. Construed broadly, an expression's conceptual base is identified as its maximal scope in all domains of its matrix (or all domains accessed on a given occasion). Construed more narrowly, its base is identified as the immediate scope in active domains –that is, the portion put 'onstage" and foregrounded as the general locus of viewing attention. Within this onstage region, attention is directed to a particular substructure, called the* ***profile****.* [13] p. 66 (bolds in the original)

Langacker illustrates the concept of profile using the week example. The concept of week is a seven-day cycle, which is the *base* and may be *profiled* in different segments called Monday, Tuesday, Wednesday, etc.

In our example, one conceptual base is the computer system and the profile may be a process or a data flow –a particular substructure– or an object in another profile. That is, the same conceptual base may be considered in terms of different profiles: data flows and processes or objects, for example.

Both ways of categorizing the computer system are different types of conceptual integrations or blends of mental spaces. The concept of mental space refers to partial cognitive structures that emerge when we think and talk *'allowing a fine-grained partitioning of our discourse and knowledge structures'.* [6] A conceptual integration or blend [7] [10] is an operation that could be applied to a couple of input spaces, which gives as a result a blended space or blend. The blend receives a partial structure from both input spaces but has an emergent structure of its own.

On the other hand, a conceptual base such as a business process, may be profiled in terms of tasks, decision points, etc., or may be also profiled as use cases, that is, subsets of the business process in which some actors –users– interact with software components in order to achieve a goal.

3 The Essential Role of Stories

The same way that use cases are based on some cognitive processes, use stories are also based on an important cognitive concept: that of stories. There is an extensive literature about the significant role of stories in cognitive sciences. Even business reviews like Harvard Business Review have published many articles about stories [4] [17] in the last years. The reason for not having paid enough attention to stories reflects prejudices about literature and its non scientific nature.

In the mid-nineties, Turner [20] and Fauconnier have developed some observant theories about stories and mental spaces. In particular, Turner has illustrated how an aspect of our mind –the literary side– provides us a knowledge that sometimes lacks a more scientific discipline.

> *The literary mind is not a separate kind of mind. It is our mind. The literary mind is the fundamental mind. Although cognitive science is associated with mechanical technologies like robots and computer instruments that seem unliterary, the central issues for cognitive science are in fact the issues of the literary mind.*
> ***Story*** *is a basic principle of mind. Most of our experience, our knowledge, and our thinking is organized as stories. The mental scope of story is magnified by* ***projection*** *–one story helps us make sense of another. The projection of one story onto another is* ***parable****, a basic cognitive principle that shows up everywhere, from simple actions like telling time to complex literary creations like Proust's* ***A la recherche du temps perdu.*** [20] p. V (bolds are italics in the original)

Turner continues telling us that his book is an attempt to show how wrong the common view is and to replace it with a view of the mind that is more scientific, more accurate, more inclusive, and more interesting, a view that no longer misrepresents everyday thought and action as divorced from the literary mind.

We constantly witness at stories that involve actors engaged in bodily actions, that is, small stories of events in space: the wind blows clouds through the sky, a child throws a ball, a girl pours water from a glass.

When we watch someone sitting down into a chair, we see what physics cannot recognize: an animate agent performing an intentional act involving basic human-scale categories of events like *sitting* and objects like *chair*. So what we, as human, additionally see is the *intention* of sitting in the animate agent.

In Requirements Engineering, the same idea of stories has been resumed in the agile movement, to the point that Leffingwell ([14] p. 56) points out that "the teams backlog consists of all the work items the team has identified. In the meta-model, we generically call these work items stories (some call them backlogs or backlog items)". The author, before using the concept of user stories defines generically *stories* as a work item.

These basic ideas of stories are resumed in the concept of user stories. A user story –used for requirements elicitation in agile and lean methods– is a particular way of describing functionality that will be valuable to either a user, developer or purchaser of a system or software. User stories are composed, in agile methods, of three aspects [2] p. 4:

- a **written description** of the story used for planning and as a reminder
- **conversations** about the story that serve to flesh out the details of the story
- **tests** that convey and document details and that can be used to determine when a story is complete

These three aspects are also called Card, Conversation, and Confirmation. The Card may be the most visible manifestation of a user story, but it is not the most important. It represents customer requirements rather than document them. This is the suitable way to think about user stories: while the card may contain the text of the story, the details are worked out in the Conversation and recorded in the Confirmation.

It is important to point out that conversations are not a detailed specification of the user story, but are merely general points that act as reminders in order to later resume the details of the story.

Conversation represents a discussion between the development team, customer, product owner, and other stakeholders, which is necessary to determine the more detailed behavior required to implement the intent. In other words, the card also represents a "promise for a conversation" about the intent. [15], p. 5.

According to a proposed template to create user stories, each story can be written in the following format:

I as a (*role*) want (*function*) so that (*business value*)

In the previous section we have seen that an expression selects a certain body of conceptual content: the conceptual base. But attention may be directed to a particular substructure, the profile.

4 Basic Level Candidate Elements

In this section we intend to show why use cases and user stories are at a basic level of categorization. We may consider three types of scenes when viewing a system: the system itself, viewed from an *external perspective*; the classical *internal perspective*, used in the old top-down method (on which ovals correspond to subprocesses) or in modern views based on UML views and the *scenario perspective*, used on use cases and user stories perspectives.

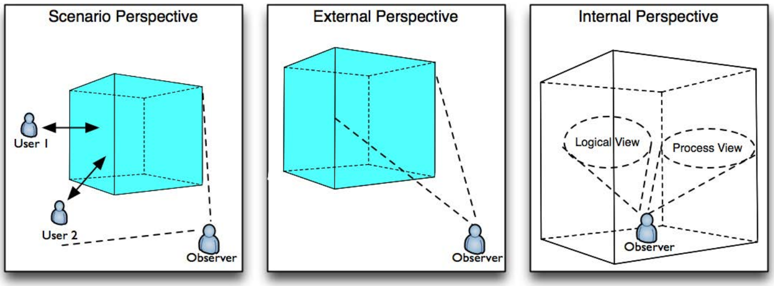


Fig. 3. Two different scenes: the enterprise and two views of the system

In classical approaches –as the top-down or previous– the external perspective represents the system such as called by its name. In Langacker's terms, we are ***focusing***– what we choose to look at– on the system from an external ***perspective*** –where we view it from.

But while in classical methods we started doing a zooming in –going directly to an internal perspective– to see the components of the system, in modern approaches –use cases and user stories– we do a zooming out to cover the users involved. This makes both pieces of elicitation essential elements in requirements engineering, as they are the equivalent of basic-level categories in cognitive semantics.

The question would be: why use cases and user stories are the basic-level categories in requirements engineering? Because they are connected to bodily motor schema: those of interacting with machines, devices and particularly, digital devices. We can easily imagine interactions with pieces of software but not with abstract subprocesses that constitute the whole system we are going to implement. In use cases and user stories somebody –the actor or the user– is the agent that triggers processes in order to obtain some valuable results.

They essentially decrease the complexity in terms of abstraction, as when associated to bodily experience, there is a more direct visualization of requirements in terms of previous similar digital devices. Another argument in favor of considering use cases and user stories as such basic-level categories is the generalized use of them in most classical, agile and lean developments.

Another important reason is the direct communication obtained with final users and the ease with which they understand the goal of such stories.

Even if both use cases and user stories may be defined as a scenario perspective, there are some differences. The main is the profile, the particular substructure to which is directed the attention: while in user stories we pay attention to the *user* and his intention (goal or business value) such as can be seen in Fig. 4, in use cases the attention is directed to the *technical structure specification*.

Sometimes we say that user stories are *business* directed but use cases are *system* directed. In fact, some authors define two types of use cases: Fowler, for example, recognizes that a *system use case* is an interaction with the Software, whereas a *business use case* discusses how a business responds to a customer or an event [8], p. 103.

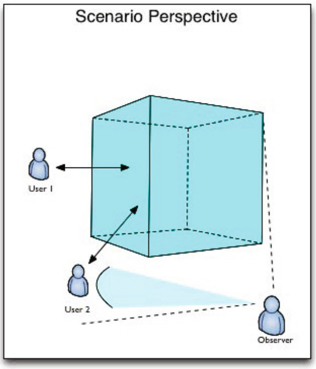


Fig. 4. In user stories the attention is directed to the user

5 Use Cases vs. User Stories

It would be interesting to consider aspects of both constructs. Leaving aside the different profiles between user stories and use cases (paying attention to the user or to the technical aspects) the other major difference is the degree of specification. When zooming in a user story, we may find some points that are not stated in a procedural way. In [5] we have some stories and conversations. One of them is the story:

> *As a Creator, I want to upload a video from my local machine so that any users can view it.*

Conversation is an open dialogue between all the team members and the client. Anyone can raise questions, ask for things to be clarified, and the answers can be recorded down as bullet points for later reference.

- *The "Upload" button will be a persistent item on every page of the site.*
- *Videos must not be larger than 100MB, or more than 10 minutes long.*

- *File formats can include .flv, .mov, .mp4, .avi, and .mpg.*
- *Upload progress will be shown in real time.*

As we can observe, these are points to be remembered, but are not a specification in natural language. They are part of a collective memory: team members and user stories.

On the other side, use cases have an associated structured template where to include a specification. In the template there are sections as: use case title, primary actor, level, precondition, postcondition, main success scenario, extensions, etc.

In this differentiation between user stories' and use cases' internals, the intentions of the team members are critical , as we can see in Fig. 5. A collaborative organization –such as in agile or lean teams– allows some decisions and specifications to be deferred until the moment they will be implemented. It is the case of a just-in-time implementation, considering each of the points of the conversation.

Indeed, in use cases the team group has decided to postpone the implementation to a later time, for different team members, as it usually occurs with more classical development processes. And in such a case the right decision is to specify the use cases in order to get the right implementation. When zooming in a user story we find some comments in the form of points to consider, and when zooming in use cases we find structured specifications in a procedural mode.

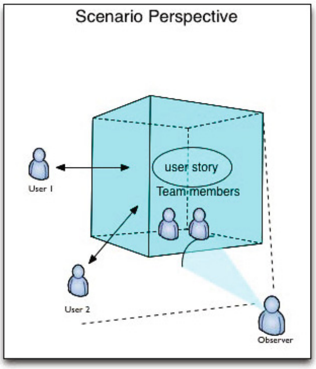


Fig. 5. The intentions of the team members are critical

Even if we limit –in this paper– the scope of use cases to agile and lean methods, we must remember that use cases were incorporated in the Rational Unified Process, a classical development method. We must also remember that use cases are also applied to large and complex systems, as exposed by Rotem-Gal-Oz [18]. But, unlike the purpose of Rotem-Gal-Oz (*[my] paper is not about explaining why use cases should be used for enterprise requirements analysis*), our aim is clearly to explain why use cases should be used.

6 Conclusions

In the early '70 Dahl, Dijkstra and Hoare published Structured Programming [3], where the authors already stated that the only efficient way to deal with complicated systems is in a hierarchical fashion. *The dynamic system is constructed and understood in terms of high level concepts, which are in turn constructed and understood in terms of lower level concepts, and so forth* [3], p. 176. The previous year, Wirth [21] had published a paper where he proposed that *the program is gradually developed in a sequence of refinement steps.* This is the essence of what was later named the top-down method.

In the following years, this approach was rapidly generalized to design and analysis, in what was called *structured design and analysis.* The problem with the approach is that it was not scalable to the analysis and, in particular, to zooming in from the higher level of abstraction –the system or software– to an intermediate level, as stablished by Yourdon when he points out the "analysis paralysis,"the "six analyst" phenomenon, or the "arbitrary physical partitioning".[22].

In the early '90 was formulated the concept of use case and in these same years appears the agile movement, which coined the concept of user stories. The long and strong history of these constructs –twenty years for use cases– is an empirical evidence of their success and that they are at the right level of abstraction: basic level.

A cognitive point of view about two of the most frequently used constructs of Requirements Engineering shows that this popularity is due to certain cognitive foundations. One important aspect of user stories and use cases is that they both are packages of requirements determining a useful purpose for the user. The other important point we have seen here is that both are at an intermediate level of abstraction, corresponding to what has been named basic-level categories in cognitive sciences.

The important fact is that basic-level categories are closely linked to motor capacity, which makes them immediately understandable: the bodily motor experiences are those of interacting with machines and devices, so there is a direct projection from the bodily –concrete– experience to the requirement artifacts we are defining. If we zoom in –in terms of a classical top-down method– to more details, we get complex sets of isolated requirements that need to be packaged into meaningful units that help to give them some meaning. But zooming in –in terms of user stories– we get conversations and the confirmation (test case) or, in terms of use cases, a semi-formal specification, depending on the applied profile.

This is the reason of their popularity and the long and solid life they have maintained through many years in agile and lean methods. Above these constructs may be some abstract components, derived from the aggregation of user stories (or use cases) that can accommodate to more general categories. The top-down approach called these abstract components subsystems or processes in general. Such components may be rebuild from use stories or use cases using a zooming out movement (Yourdon proposed to speak of middle-out method, from which we move bottom-up) and categorizing them as bigger groups, but they can not be *discovered* as the top-down method sometimes intended to do.

As future work, we plan to evaluate the possibility of extending the concept of use case –as based on human body schemas– to more formal constructs, such as used in complex systems like operating systems, mission critical systems, etc. Operating systems, for example, are based on concepts like tasks, jobs or commands. Considered in detail, these concepts are metaphorical extensions of human activities and we may investigate the concrete activities which constitute the source mental space of such metaphors. The immediate observation we can make is that tasks are most likely abstract views of use cases, in which actors correspond to events that trigger the tasks, the task itself correspond to a use case and the result of the execution is common to both constructs.

As Turner says: *in our small stories, we distinguish objects from events, objects from other objects, and events from other events. We categorize some objects as belonging to the category* ***person*** *and other objects as belonging to the category* ***chair***. [20] pp. 14-15 (bolds are italics in the original). That is, what is involved in stories, generically, are objects and events.

Acknowledgements. The author is grateful to Mauricio Milchberg for his revision of the manuscript and valuable comments.

References

1. Allwood, J., Gärdenfors, P. (eds.): Cognitive Semantics: Meaning and Cognition. John Benjamins Publishing Company, Amsterdam (1999)
2. Cohn, M.: User Stories Applied: For Agile Software Development. Addison-Wesley Professional (2004)
3. Dahl, O.-J., Dijkstra, E.W., Hoare, C.A.R.: Structured Programming. especially. Academic Press (1972)
4. Duarte, N.: Structure Your Presentation Like a Story. HBR (2012), `http://blogs.hbr.org/cs/2012/10/structure_your_presentation_li.html`
5. Dennis, S.: How to write meaningful User Stories (2010), `http://www.subcide.com/articles/how-to-write-meaningful-user-stories/`
6. Fauconnier, G.: Mappings in thought and language. Cambridge University Press, Cambridge (1997)
7. Fauconnier, G., Turner, M.: The Way We Think: Conceptual Blending and the Minds Hidden Complexities. Basic Books (2002)
8. Fowler, M.: UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd edn. (2003)
9. Imaz, M.: Abstracción y Conceptualización: Un Enfoque Cognitivo. 40 JAIIO, Jornadas Argentinas de Informática, 29 de Agosto al 2 de Septiembre de 2011, Córdoba, Argentina. Anales 40 JAIIO /CD40JAIIO/T2011/ASSE/603.pdf (2011)
10. Imaz, M., Benyon, D.: Designing with Blends. MIT Press (2007)
11. Kramer, J.: Is Abstraction the Key to Computing? Communications of the ACM 50(4), 37–42 (2007)
12. Lakoff, G.: Women, Fire, and Dangerous Things: What Categories Reveal about the Mind. University of Chicago Press, Chicago (1987)
13. Langacker, R.: Cognitive Grammar: A Basic Introduction. Oxford University Press, Inc., New York (2008)

14. Leffingwell, D.: Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise. Addison-Wesley Professional (2011)
15. Leffingwell, D., Behrens, P.: A User Story Primer. Whitepaper (2003), http://scalingsoftwareagilityblog.com/wp-content/uploads/2010/01/user-story-primer_1.pdf
16. Margolis, E., Laurence, S. (eds.): Concepts: Core Readings. MIT, Cambridge (1999)
17. McKee, R., Fryer, B.: Storytelling That Moves People HBR (2013), http://hbr.org/web/special-collections/insight/communication/storytelling-that-moves-people
18. Eotem-Gal-Oz, A.: Methodology for developing Use Cases for large systems? (2004), http://www.rgoarchitects.com/blog/content/binary/uc.pdf Methodology for developing Use Cases for large systems
19. Talmy, L.: Toward a Cognitive Semantics. Concept structuring systems, vol. 1. MIT Press, Cambridge (2000)
20. Turner, M.: The Literary Mind. Oxford University Press, Oxford (1996)
21. Wirth, N.: Program Development with Stepwise Refinement. Communications of the ACM 14(4), 221–227 (1971)
22. Yourdon, E.: Modern Structured Analysis. Prentice Hall, Englewood Cliffs (1989)

Using Distributed Cognition Theory for Analyzing the Deployment Architecture Process

Naomi Unkelos-Shpigel and Irit Hadar

Information Systems Department, University of Haifa
Carmel Mountain 31905, Haifa, Israel
{naomiu,hadari}@is.haifa.ac.il

Abstract. Deployment architecture is an important part of the software development lifecycle. Our preliminary research indicates that the process of constructing the deployment architecture holds several challenges which, if not properly met, may hinder the success of the project and result in low customer satisfaction. In this ongoing research, we analyze this process in different firms, in an attempt to understand its challenges towards proposing strategies for its improvement. To this end, we collected data via 25 questionnaires at a global IT firm, and 12 in-depth interviews with architects from 10 firms. Analyzing the data qualitatively, through the lens of distributed cognition theory, helped us to understand the structure and flow of this multiple-stakeholders process and identify common potential pain points and challenges that need to be addressed.

Keywords: deployment architecture, distributed cognition, qualitative research.

1 Introduction

Deployment architecture is an important part of the software development lifecycle. Its purpose is to bridge the gap between the requirements of the capabilities and the solution delivered and installed at the customer`s site [1]. It is defined as "allocation of the system's software components (and connectors) to its hardware hosts" [2, p.1], and highly influences the quality of service (QoS) [2].

Since the deployment solution involves adjusting the product to the customer`s needs and environment, it has a major effect on customer satisfaction of the product [3]. Several empirical studies have examined the deployment process, mostly focusing on creating the deployment architecture and understanding its requirements [4]. Deployment architecture concerns that are not widely inquired include the challenges along different phases of the deployment process and the different tools and support the service architect (who creates the architecture, hereafter referred to as the architect) can rely upon. These concerns may influence the decisions made by the architect while constructing the deployment architecture solution and specifically, the components selected to be part of this solution.

The objective of this ongoing research is to enhance the deployment architecture process, resulting in better solutions and higher customer satisfaction. Accordingly, as a first step, our empirical study aims at identifying the challenges and difficulties that

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 186–191, 2013.

may hinder its success. To this end, we study the flow of the architecture deployment process, its different participants, and their decision-making and reasoning processes.

Some attention has been given in the research areas of software and IS engineering to cognitive processes related to different activities of development. For example, a survey on cognitive studies in the context of software analysis and design can be found in [6]. However, we could not find empirical studies investigating cognitive processes of software architects in general and of deployment architects in particular. In this study, we make a first step towards investigating the construction of deployment architecture from a cognitive point of view. Considering that the creation of deployment architecture involves several different sources – both human and non-human – we recruited for our analysis the theory of distributed cognition.

Distributed cognition separates a system, e.g., a collection of people and tools working together to achieve a goal, to subsystems, referring to the cognition of each subsystem in the process as a separated unit [5]. It addresses the concerns of the different subsystems, their interaction, conflicts and dilemmas that stem from their different points of views. Distributed cognition further proposes strategies to improve the systems and its processes based on the identified conflicts This theory fits very well to our objective since the creation of the deployment architecture solution is a complex process, which involves different stakeholders and various tools and artifacts. Addressing each of these as a unique subsystem, with its own cognition and perception of the process, can provide us a better understanding of the process, its weaknesses, and accordingly, opportunities for its improvement.

2 Distributed Cognition

Distributive cognition is based on the principle that "cognitive processes are socially distributed across the members of a group" [7]. It extends the reach of what is considered cognitive beyond the individual, and relates to the interactions between people, materials and resources. The theory looks for a broad class of cognitive events and does not expect all such events to occur within the physical boundary of an individual, in contrast with other popular cognitive theories [7]. Social organizations are considered forms of cognitive architecture. A system can dynamically configure itself, bringing subsystems into coordination to accomplish various goals. A cognitive process is delimited by the functional relationships among the elements that participate in it, rather than by the spatial collocation of the elements. The context of an activity determines the way information flows through a group. Memory is distributed among various subsystems, and each unit (subsystem) has its own cognition, according to the actions it performs; this can lead to some contradictions in the perceptions of different units [8]. The study of cognition is not separable from the study of culture; culture provides us with intellectual tools that enable accomplishing things that could otherwise not have been completed [7].

The study of distributed cognition is usually done ethnographically, focusing on the meaning of the systems and the meaning of the words used during the process [7]. In contrast to classic cognitive theories, the focus here is not only on the knowledge of people, but also on how they use what they know in order to do what they do. The theory claims that cognitive activity is constructed from both internal and external resources and that the meanings of actions are grounded in the context of an activity. This theory can be used to improve the design of working processes [8].

In the domain of IS engineering, this theory has been used mainly in HCI research [7,9]. Nillson et al. [9] used distributed cognition to analyze human-machine interaction, and the term socio-technical system, which includes the people carrying out a task and the artifacts they use. This created a new perspective on the structure and role of the subsystem in the process; a subsystem can include both human and non-human parts, and interact with other such subsystems [9].

3 Empirical Study

3.1 Method and Settings

The main objective of our study is to identify the challenges and difficulties that may hinder the quality of the deployment architecture solution. To this end, we conduct a qualitative research, where the investigator does not have a pre-defined theory about the environment inspected, but rather uses techniques and tools to explore and discover phenomena [10]. We use tools and methods based on the grounded theory methodology [10]; data is collected from the field and inductively analyzed.

We initiated our research at a global, large-scale IT firm by a pilot of interviews, for a first acquaintance with the firm and its deployment division, followed by distributing an open-questions questionnaire to deployment architects. The questionnaire was aimed to achieve a preliminary identification of the major concerns of the process and opportunities for improvements, as perceived by architects. In order to enrich our data beyond this single case study [11] we collected data via in-depth interviews in additional firms. All together, ten global firms – six hi-tech firms, two communication firms and two shipping firms – were included in the research so far. Only global firms that regularly execute major deployment projects were included in the research. The interviewees were experienced deployment architects, working with customers as well as with internal divisions responsible for defining products. In what follows we present preliminary results obtained thus far in this research.

3.2 Analysis and Findings

Using an inductive qualitative data analysis method [10], we divided the text from the interviews transcriptions and questionnaires to statements and classified them to categories of challenges within the process. All together, 80 relevant statements were identified and classified. Next, we mapped the emergent categories to the unit of analysis as defined by the distributed cognition theory. When applying grounded theory, consideration of literature is allowed for guiding data analysis [12].

The first and most profound principle of distributed cognition is to determine the unit of analysis, in which cognition will be examined. Since we explore the process of deployment architecture, we defined the unit of analysis to include all stakeholders of this process, as well as the resources and systems used. The different subsystems included in our unit of analysis, based on the data collected, are described in figure 1.

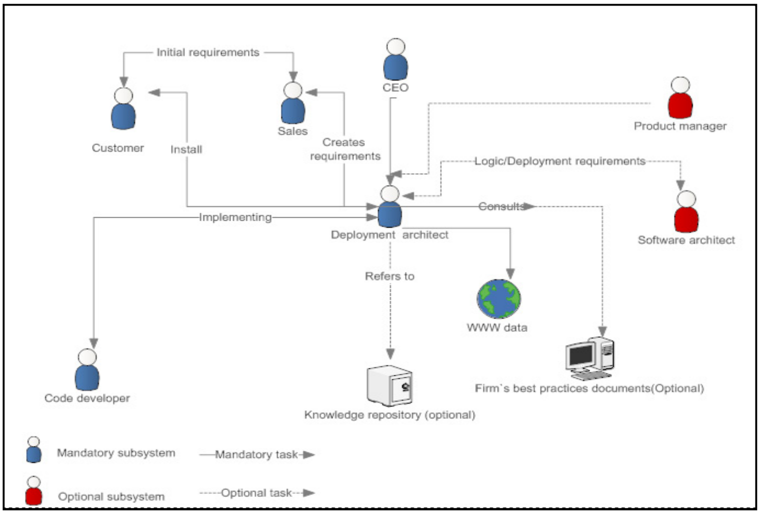


Fig. 1. Subsystems in the deployment architecture process

In what follows we present and briefly explain each category and its related challenges. Due to space limitations, we will not present here exemplary quotes.

Memory Distributed among Subsystems: Having the memory distributed among subsystems, as explained by the distributed cognition theory, results in different subsystems having different view of the process [7]. This often leads to conflicts and contradictions between subsystems.

Challenge: Having organizational memory distributed among different subsystems causes conflicts during the process. It leads each subsystem to focus on its own interests and priorities. For example, architects often find themselves constrained to promises given by salespersons, which in some cases are not provided by the product. This happens usually because the communication between these two subsystems occurs at a late stage, after all the requirements of the product have been established. Such cases result in either high costs of patching solutions, or in failure to deliver.
This challenge is typical to firms not using a knowledge repository.

Internal and External Representations of Concepts/Artifacts: Each subsystem has its individual cognitive processes based on its own knowledge and point of view, in addition to the information communicated by other subsystems [7]. This results in interpretation of an external representation of information (e.g., requirements document) to an internal, often different representation of this information, which affects the action taken by the subsystem.

Challenge: The difference between internal and external representations often causes misunderstandings during the process. For example, when different stakeholders refer to the same item using different terminology. As a result, major issues regarding product features may be misinterpreted and thus not delivered as intended. This challenge was found in all the firms in our study except for one. In this firm, a separate team was responsible for creating an overall taxonomy for the process; all stakeholders were instructed to use these unified terms.

Interaction among Subsystems: The different subsystems communicate with each other in order to complete the process successfully [5,p.251]. However, this dependency on communication might lead to tension and conflicts between stakeholders.

Challenge: We encountered in several cases that the communication between code developers and project management is weak or non-existent. Thus, managers are often not informed about problems related to product development. Only later, during the deployment, the product is found to be not eligible to be deployed at customer`s site: Another example of missing communication between subsystems is related to lack of information about the quality of different components of the product, which failed tests in the context of other stakeholders. This problem was mentioned by many of the architects; we found no solution for this problem in any of the firm we studied.

Interface with Tools: Interaction between human and non-human subsystems and the effect of the latter on the cognitive processes of the former is another focus of distributed cognition. As illustrated in figure 1, the deployment architect uses other subsystems, which are automated tools such as remotely accessible knowledge repositories or the Internet. This eliminates the need in spatial proximity between subsystems, and implies that the context of action (the purpose of using the tool) denotes its meaning [7].

Challenge: Due to the differences in focus, different stakeholders use different tools to create and manage information. While the architect should be able to see all different views, no single tool can display all the views of all the stakeholders at the same time. This causes the architect to waste additional time on looking for data in multiple tools and systems, viewing only fragments of the information at a time, which makes it difficult to see the bigger picture. Moreover, some tools are over complex for their purpose, leading stakeholders to work with simpler tools (e.g., excel), which are sufficient for them but do not provide the wide perspective needed for the architect. This challenge is typical to firms, which do not have a holistic knowledge management solution, with a knowledge repository and appropriate tools for each stakeholder.

Cultural Environment: Culture influences cognitive processes both within and between subsystems [7].

Challenge: A good example, which we witnessed in several American firms, although global in nature, is the tendency to neglect addressing international regulations. Firms failing to consider these factors, usually when management does not put enough emphasis on the matter and due to the natural focus on the local market, often face severe problems when deploying their product in countries where unique regulations on this type of product are defined. This challenge was mostly found in firms based in countries with rigid regulations. Firms addressing this concern as part of their working process, instruct the participants of the deployment process to address international regulations at an early stage, and construct the solution accordingly.

4 Conclusion

This paper demonstrated the analysis of the deployment architecture process through the lens of distributed cognition. Using this theory we were able to refer to each of the stakeholders and tools as a separate subsystem, having its own cognition and interactions with other subsystems. We used this perspective to analyze the data we collected on the deployment architecture process, and to identify the challenges deployment architects face and their sources.

In contrast to typical empirical studies on distributed cognition, based mostly on observations, we based our findings on statements taken from interviews and questionnaires. However, since we used two different data collection tools – questionnaires and interviews, and different data sources spreading over 10 firms from different domains, we believe this limitation is mitigated, at least to some extent, by this triangulation. Having said that, it should be noted that this is still an ongoing study, and that the findings presented here are merely preliminary. We plan to refine and validate our findings via additional interviews and observations. Next, we will develop strategies for improving the deployment architecture process based on our findings and on the principles of the distributed cognition theory.

References

[1] Carzaniga, A., Fuggetta, A., Hall, R.S., Heimbigner, D., Van Der Hoek, A., Wolf, A.L.: A characterization framework for software deployment technologies.Technical Report. Colorado State Univ. For Collins Dept. of Computer Science (1998)

[2] Medvidovic, N., Malek, S.: Software deployment architecture and quality-of-service in pervasive environments. In: International Workshop on Engineering of Software Services for Pervasive Environments, in Conj. with the 6th ESEC/FSE Joint Meeting. ACM (2007)

[3] Mockus A., Zhang P. and Li L.: Predictors of customer perceived software quality. In: Proceedings of the International Conference on Software Engineering (ICSE 2005), pp. 225–233. ACM (2005)

[4] Adam, S., Doerr, J.: On the notion of determining system adequacy by analyzing the traceability of quality. In: Proceedings of Workshops and Doctoral Consortium (2007)

[5] Hutchins, E.: Cognition in the Wild. MIT Press, Cambridge (1995)

[6] Hadar, I.: When Intuition and Logic Clash: The Case of the Object Oriented Paradigm. Science of Computer Programming (in press) http://dx.doi.org/10.1016/j.scico.2012.10.006

[7] Hollan, J., Hutchins, E., Kirsh, D.: Distributed cognition: toward a new foundation for human-computer interaction research. ACM Transactions on Computer-Human Interaction (TOCHI) 7(2), 174–196 (2000)

[8] Ackerman, M.S., Halverson, C.: Considering an organization's memory. In: Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work, pp. 39–48. ACM (1998)

[9] Nilsson, M., Laere, J.V., Susi, T., Ziemke, T.: Information fusion in practice: A distributed cognition perspective on the active role of users. Information Fusion 13(1), 60–78 (2012)

[10] Strauss, A., Corbin, J.: Basics of qualitative research grounded theory procedures and techniques. Sage Publications, Inc. (1990)

[11] Walsham, G.: Interpretive Case Studies in IsResearch: Nature and Method. European Journal of Information Systems 4(2), 74–81 (1995a)

[12] Suddaby, R.: From the editors: What grounded theory is not. Academy of Management Journal 49(4), 633–642 (2006)

Cognitive Principles to Support Information Requirements Agility

Jeffrey Parsons[1] and Yair Wand[2]

[1] Faculty of Business Administration,
Memorial University of Newfoundland St. John's, Canada
[2] Sauder School of Business,
The University of British Columbia Vancouver, Canada
jeffreyp@mun.ca, yair.wand@ubc.ca

Abstract. Despite the growing interest in agile information systems development approaches, we contend that existing approaches retain traditional assumptions about the structure of information, assumptions that inhibit agile responses to emerging and evolving information requirements. We suggest an approach, based on cognitive principles, to model information requirements by separating conceptual views of data from logical models, allowing the former to be changed without requiring changes to the latter.

1 Introduction

Traditional approaches to information systems engineering were based on some version of the waterfall model, in which development is undertaken in well-defined stages, with each being completed before the next stage started. Such a process entails developing a comprehensive set of system requirements early in the process. This approach assumes the stakeholders of an information system are a well-defined group, whose functional requirements are known and can be elicited systematically [1, 2].

In addition, these approaches make a number of (usually implicit) assumptions about data requirements, including: (1) the sources of the data are well-understood; (2) there is some centralized management of data (a global conceptual schema), possibly shared by several applications; and (3) the information structure (conceptual and logical design) is relatively stable over time.

In view of the limitations of the waterfall model, the focus in information systems engineering has shifted to approaches that emphasize modularity and agility. In particular, requirements emerge and are refined through several iterations, during which additional functionality is added incrementally based on interaction with users and other stakeholders and the availability of a prototype or partially implemented system.

Notwithstanding these developments, we contend that agile development approaches retain implicit assumptions about the structure and stability of information requirements characteristic of waterfall methods. Specifically, although agile methods recognize that functionality and user interfaces might change often across iterations, the structure of information requirements (and therefore, of data in the

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 192–197, 2013.

database) is assumed to be relatively stable. There are two reasons why the data aspects of applications might be less volatile than the processing and interface aspects. First, data reflect a view of the application domain in terms of a set of concepts and relationships among them. While the required functionality might change relatively easily, it is usually implicitly assumed that stakeholders' views of the domain are quite stable over time. Second, functionality might often be changed or added without a need to "retrofit" the developed system, whereas changes in data structures might force changes to the implementation of many functional components. It follows that, during the development process, one should not expect much agility with respect to the structure of the information requirements for the system.

We claim, however, that modern application environments do require more flexibility and agility to accommodate changes to the information itself. The reasons for such changes include: (1) the emergence of new (not originally anticipated, and often external) sources of data (e.g. on the Internet), the structure of which might not match the conceptual model elicited during initial requirements determination; (2) the appearance of new users or stakeholders who might have different domain views that were not captured during initial requirements determination; and (3) the discovery of new information requirements that were not anticipated earlier. Such changes would typically call for changes in data structures and consequently the functionality already implemented might need to be re-implemented.

Information requirements are manifested via the structure of the data in the application. Databases were introduced in order to decouple issues of access to data from the physical structures used to store data, making it possible to change physical data structures without affecting the implemented functionality or requiring changes to the mechanisms used to store and retrieve data. However, this independence does not extend to changes to logical or conceptual data structures. Specifically, logical data structures often are designed to reflect specific views (as embedded in conceptual data models). Thus, when new or different user views emerge, data models need to be changed accordingly, and the functionality that has been already implemented based on these structures might need to be changed as well.

In this position paper, we suggest an approach, based on cognitive principles, that enables flexibility in supporting different views of a domain, without having to re-implement existing functionalities. If this can be done, then agility can be extended to changes in users' views of a domain. Thus, an earlier commitment to a specific user view will not constrain the possible changes that can be accommodated in later development cycles. As user views are manifested in the form of information requirements, we believe our approach will result in agility with respect to these requirements.

2 An Example

We begin with an example demonstrating how information requirements changes may cause changes of data structure (which in turn might affect processing elements of a system). We first present a specific view and the related data structure design. We then discuss changes to the view arising from changing application requirements, and how these might impact the database design and accordingly query implementation. Later we revisit the example to show how such view changes can be accommodated in our proposed approach without impacting the implementation of functionality.

Consider a university information system that manages student data. A graduate school is responsible for its thesis-based graduate students. It manages personal student information. In addition, each student has a supervisor, and the system needs to keep track of the assignment of students to supervisors.

In this highly simplified system, the information requirements can be captured in a relational data model, consisting of a single table:

STUDENT(<u>Student_ID</u>, Name, Address, Supervisor).

Note that, if it is necessary to keep track of supervisor information, this can be extended by adding a table with supervisor information. Suppose further that a basic reporting requirement is to list the name of students from a particular location, X, and their supervisors. This can be satisfied with the following SQL query:

```
select Name, Address, Supervisor
  from student where Address=X
```

Now suppose new requirements arise from the introduction of a course-based graduate program (in addition to the existing thesis-based program). It is necessary to maintain the same personal information about students in this new program. However, these students do not have a supervisor; instead, they participate in a designated major. The existing logical data model does not accommodate this new reality very well. It is possible to extend the definition of the STUDENT relation by adding a \`major\` attribute and allowing null values for both ‘major’ and ‘supervisor’. However, given the resulting ambiguity in interpreting the meaning of null (‘not applicable’ versus ‘unknown’), a more appropriate solution is to create two new relations capturing the information unique to each specific subclass of student (thesis and course-based)[1]:

STUDENT(<u>Student_ID</u>, Name, Address)
THESISSTUDENT(<u>Student_ID</u>, Supervisor)
COURSESTUDENT(<u>Student_ID</u>, Major)

Given this change in the logical data model resulting from the evolving requirements (manifested by a new view of the kinds of entities about which information must be maintained), the existing query to extract student information and supervisor names for students with a particular address will no longer work. Instead, it will be necessary to join the STUDENT and THESISSTUDENT tables to answer this query. Thus, any functionality relying on the original query will not be supported until the query is rewritten. Moreover, it will be necessary to repopulate the tables in the new design based on the changes made. In this case, both the original database structure and the queries can be considered ***brittle***, as they might not work (without modifications) when the requirements of the application domain change. Note also that the queries cannot be written or adjusted to the new requirements without detailed knowledge of the specific schema chosen.

[1] It is possible to create only two tables for the new requirements and maintain the names and addresses of students in the tables for the particular type of student (thesis vs. course-based). In that case, it will be more complicated to query information common to all students.

To address this brittleness, we introduce cognitive principles that can guide the design of flexible data structures to naturally accommodate changing views of the domain without requiring changes to the queries that support information requirements.

3 Cognitive Principles

We aim to identify principles that will enable changes to the conceptual views of the application domain without requiring changes to the underlying data structures. To identify such principles we turn to cognitive science, specifically, to ***classification theory*** [8,9], which deals with how humans form concepts based on their experiences. In particular, we focus on two cognitive principles of classification: (1) instances possess properties and exist independent of how they are classified; (2) classes do not exist independent of human cognition, but instead are abstractions of useful similarities formed based on observed properties of instances. Different abstractions are formed to reflect different views (reflecting different uses).

We suggest these principles be applied to information systems design by using the following design rule [4]:

Design Rule 1: Instances in an information system should be stored with their properties, independent of any specific classification.

To understand the significance of this rule, note that the common approach places data in pre-determined "containers", which usually reflect a given view (i.e., a particular set of classes). The most common type implementation of a container is the relation in the relational model. The proposed rule actually means that database design will not be driven by (or reflect) any particular user view, except that a view may indicate the relevant properties of instances.

The above design rule does not deny the importance of classes in information systems engineering. Classes still have a vital role in requirements analysis, system design, and data access. Rather, it considers the issue of which classes are relevant to information requirements as completely separate from the mechanisms to store data. Hence, we term this as applied to design ***class independence***.

As user views as well as accessing and entering information will still be defined in terms of concepts or classes, we add a second design rule:

Design Rule 2: Mechanisms must be provided to define classes in terms of properties and to retrieve or access instances of a given class.

Given that, in our approach the database stores only instances. Retrieving all instances of a class means identifying all instances possessing a given set of properties.

These design rules imply that classes can be defined and modified, and data accessed in terms of classes, without a need to change the way instances are stored. Classes thus become completely abstract, and do not serve as "containers" for data.

Consider now how class independence can support flexibility in domain views. This requires first that instances be related to classes. Such relationships reflect how classes are defined. The simplest way to do this is the classic approach where a class is defined in terms of a set of properties. Any instance possessing this set will be considered an instance of the class. A class-independent implementation will require operations to support identifying the instances of a class and the classes to which an instance belongs.

4 Implementing the Cognitive Principles to Support Information Requirements Agility

Consider again the student example (Section 2). We can envision a domain of instances possessing properties as the foundation for the logical design of data. There are various ways to implement this idea. For consistency with the example above, we assume a relational database implementation.[2] One way to realize the instance-based approach is to have a separate (binary) table representing each property of interest. In our example, given the initial information requirements, we could have three tables:

NAME(Student_ID, Name)
ADDRESS(Student_ID, Address)
SUPERVISOR(Student_ID, Supervisor).

As explained above, in the instance-based we need a mechanism to define classes in terms of properties. For example the class STUDENT can be defined as:

STUDENT = {Name, Address, Supervisor}.[3]

For the emerging requirement to accommodate students in course-based programs, all that is needed is to add a new property for Major to each relevant instance (in our implementation example this will entail adding a binary table where each row represents an instance possessing the new property). In addition, a class structure to reflect the new view can be defined ***without making any changes to the underlying data***:

THESISSTUDENT = {Name, Address, Supervisor}
COURSESTUDENT = {Name, Address, Major}.

This structure is purely a view over the attributes. The base (binary) tables do not change as new classes are defined, existing classes are modified (their properties change), or classes are eliminated. For example, if a list of all students (both thesis and course-based) is needed, the class STUDENT = {Name, Address} can be added without any change to data structures or to other classes.

It can be readily shown that under this structure, any previously defined queries (such as to retrieve the names, addresses and supervisors of students having a particular address) will continue to work.

From this example, we see that a logical data model based on cognitive principles of classification offers considerable flexibility to accommodate changes in information requirements and to views of the domain. In contrast, models such as the relational model that embed a certain (class-based) view of the domain in the logical data model are cumbersome, requiring changes at the design level to accommodate changing requirements and/or views of the domain.

---

[2] Note that other ways of realizing this design are possible. Graph-based structures provide a natural mechanism to support domain conceptualization in terms of instances and properties. Similarly, column-oriented and key-value structures are similar to our binary table approach.

[3] Technically, STUDENT can be defined as a view through a query joining the Name, Address, and Supervisor tables on the Student_ID attribute. The identity of the student does not appear explicitly, as it is not part of a conceptual class, but rather an implementation mechanism. Such identity will exist however in the underlying data structures (e.g. relational).

5 Conclusions and Further Opportunities

Requirements in information systems engineering clearly fall within the domain of cognition. Developing an information system often begins without a full understanding of stakeholder requirements. Moreover, in modern technologies and applications, requirements cannot be expected to remain stable over time. The uncertainty and volatility surrounding requirements has been recognized in software engineering, leading to the rise of agile approaches.

However, one aspect of development has been largely ignored in the pursuit of agility – the nature of ***information*** requirements and their manifestation in database design. We contend that this is fundamentally a cognitive issue, as information requirements reflect stakeholder conceptualizations of the important phenomena of interest in the domain. We argue that the role of cognition, beyond being closely linked to users views and hence requirements, can be extended to data design principles that support changing information requirements that are now commonplace.

Here we have not addressed the actual process of eliciting users' views. Elsewhere we describe how additional cognitive principles can be used to determine user views [3,6], to evaluate alternative views [6], to guide conceptual data design [3], to integrate different views [Parsons & Wand 2003], and as underlying foundations for general information processing architecture [7].

References

1. Browne, G.J., Rogich, M.B.: An Empirical Investigation of User Requirements Elicitation: Comparing the Effectiveness of Prompting Techniques. Journal of Management Information Systems 17(4), 223–249 (2001)
2. Hickey, A.M., Davis, A.M.: A Unified Model of Requirements Elicitation. Journal of Management Information Systems 20(4), 65–84 (2004)
3. Parsons, J., Wand, Y.: Choosing Classes in Conceptual Modeling. Communications of the ACM 40(6), 63–69 (1997)
4. Parsons, J., Wand, Y.: Emancipating Instances from the Tyranny of Classes in Information Modelling. ACM Transactions on Database Systems 5(2), 228–268 (2000)
5. Parsons, J., Wand, Y.: Attribute-Based Semantic Reconciliation of Multiple Data Sources. Journal on Data Semantics 1(1), 21–47 (2003)
6. Parsons, J., Wand, Y.: Using Cognitive Principles to Guide Classification in Information Systems Modeling. MIS Quarterly 32(4), 839–868 (2008)
7. Parsons, J., Wand, Y.: Extending Classification Principles from Information Modeling to Other Disciplines. Journal of the Association for Information Systems 14(3) (2013)
8. Rosch, E.: Principles of Categorization. In: Rosch, E., Lloyd, B. (eds.) Cognition and Categorization, pp. 27–48. Erlbaum, Hillsdale (1978)
9. Smith, E.E., Medin, D.L.: Categories and Concepts. Harvard University Press (1981)

Change Patterns for Model Creation: Investigating the Role of Nesting Depth*

(Position Paper)

Barbara Weber[1], Jakob Pinggera[1], Victoria Torres[2], and Manfred Reichert[3]

[1] University of Innsbruck, Austria
{barbara.weber,jakob.pinggera}@uibk.ac.at
[2] Universitat Politècnica de València, Spain
vtorres@pros.upv.es
[3] University of Ulm, Germany
manfred.reichert@uni-ulm.de

Abstract. Process model quality has been an area of considerable research efforts. In this context, the correctness-by-construction principle of change patterns offers a promising perspective. However, using change patterns for model creation imposes a more structured way of modeling. While the process of process modeling (PPM) based on change primitives has been investigated, little is known about this process based on change patterns and factors that impact the cognitive complexity of pattern usage. Insights from the field of cognitive psychology as well as observations from a pilot study suggest that the nesting depth of the model to be created has a significant impact on cognitive complexity. This paper proposes a research design to test the impact of nesting depth on the cognitive complexity of change pattern usage in an experiment.

Keywords: Process Model Quality, Process of Process Modeling, Change Patterns, Exploratory Study, Problem Solving.

1 Introduction

Much conceptual, analytical, and empirical research has been conducted during the last decades to enhance our understanding of conceptual modeling. In particular, process models have gained significant importance due to their fundamental role for process-aware information systems. Even though it is well known that a good understanding of a process model has a direct and measurable impact on the success of any modeling initiative [1], process models display a wide range of quality problems impeding their comprehensibility and maintainability [2].

To improve process model quality, change patterns offer a promising perspective [3]. Instead of creating a process model using change primitives (e.g., add node, add edge) high-level change operations combining several change primitives are used as basic building blocks for model creation. Examples of change

* This research is supported by Austrian Science Fund (FWF): P23699-N23.

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 198–204, 2013.

patterns include the insertion and deletion of process fragments or their embedding in loops. Particularly appealing is correctness-by-construction [3], i.e., the modeling environment provides only patterns to the process modelers, which ensure that a sound process model is transformed into another sound model.

The use of change patterns implies a different way of creating process models, since correctness-by-construction imposes a structured way of modeling by enforcing block structuredness. Irrespective of whether change patterns or change primitives are used, model creation requires process modelers to construct a mental model (i.e., *internal representation*) of the requirements to be captured in the process model [4]. In a subsequent step, the mental model is mapped to the constructs provided by the modeling language creating an *external representation* of the domain [4]. While the construction of the mental model presumably remains unaffected, the use of change patterns leads to different challenges concerning pattern selection and combination for creating the external representation. In particular, process modelers might have to look several steps ahead to construct a certain process fragment, which constitutes a major difference compared to the use of change primitives, which do not impose any structural restrictions.

The process of creating process models based on change primitives has caused significant attention leading to a stream of research on the *process of process modeling* (PPM) [4–7]. This research is characterized by its focus on the formalization phase of model creation, i.e., the modeler's interactions with the modeling environment [5]. Still, little is known about the PPM when utilizing change patterns. To fill this gap, we conducted a pilot study with 16 process modelers [8], which indicated that the cognitive complexity imposed by change pattern usage is highly related to the structure of the process model to be created, in particular the nesting depth of the model. Modelers did not face any major problems when constructing simple process fragments, e.g., when inserting activities in sequences, making an activity optional, or inserting an activity in parallel. Faced with more complex control flow structures, in turn, the structural restrictions imposed by change patterns led to considerable problems (i.e., detours or incorrect models). These observations were underlined by feedback of the participants who appreciate the correctness-by-construction guarantees, but feel restricted when faced with complex control flow constructs. To further investigate these observations this paper proposes a research design to test the influence of nesting depth on the cognitive complexity of change pattern use.

2 Cognitive Foundations of Problem Solving

We consider the creation of process models to be a complex problem solving task. Problem solving has been an area of vivid research in cognitive psychology for decades. Therefore, we turn to cognitive psychology to understand the processes followed by humans when solving a problem like creating a process model.

Schemata. The human brain contains specialized regions contributing different functionality to the process of *solving complex problems*. *Long-term memory* is responsible for permanently storing information and has an essentially unlimited capacity, while in *working memory* comparing, computing and reasoning take place [9]. Although the latter is the main working area of the brain, it can store only a limited amount of information, which is forgotten after 20–30 seconds if not refreshed [10]. The question arises how information can be processed with such limited capacity. The human mind organizes information in interconnected *schemata* rather than in isolation [9]. Those schemata, stored in long-term memory, incorporate general concepts of similar situations [9]. Whenever situations similar to a schema arise, the latter is retrieved to help organizing information by creating *chunks* of information that can be processed efficiently [11]. To illustrate how chunking actually influences the understandability of process models consider a fragment with one alternative branch. Unexperienced modelers may use three chunks to store such process: one for each XOR- gateway and one for the activity. In contrast, an expert may recognize the pattern for optional activities, i.e., a schema for optional activities is present in long-time memory, allowing the storage of the entire process fragment in one working memory slot.

Problem-Solving Strategies. Novices confronted with an unfamiliar problem cannot rely on specialized problem solving strategies. Instead, an initial skeletal *plan* is formed [12]. Then, they utilize general problem solving strategies, like means-ends analysis, due to the lack of more specific strategies for the task [13]. Means-ends analysis can be described as the continual comparison of the problem's current state with the desired end product. Based on this, the next steps are selected until a satisfying solution is found [13]. After applying the constructed plan, it can be stored in long-term memory as *plan schema* [12]. For this, task-specific details are removed from the plan schema resulting in a plan schema that can be automatically applied in similar situations [14]. When confronted with a problem solving task in the future, the appropriate plan schema is selected using case-based reasoning [15]. The retrieved plan schema provides the user with structured knowledge that drives the process of solving the problem [11, 15]. Plan schemata allow experts to decide what steps to apply to end up with the desired solution [16]. If the plan schema is well developed, an expert never reaches a dead end when solving the problem [17].

Plan schemata seem important when creating process models based on change patterns since patterns cannot be combined in an arbitrary manner, especially when complex control-flow structures have to be created. If no plan schema is available on how to combine patterns to create the desired process model, modelers have to utilize means-ends analysis until a satisfying solution is found. This behavior is more likely to result in detours and decreased modeling speed. Moreover, the cognitive complexity for conducting means-ends analysis increases when confronted with complex control-flow structures like deeply nested blocks, making it more difficult to reach the correct solution. In addition, respective structures require the modeler to possess schemata to process larger chunks of information beforehand (i.e., increased need for look-ahead) to be able to model the respective fragment fast and without any detours. As a consequence, mental effort increases as well as the probability for detours.

3 Research Design

Based on the cognitive foundations in Sect. 2 we propose a research design to investigate the influence of nesting depth on the cognitive complexity of change patterns usage for model creation by means of a controlled experiment.

Subjects. As explained in Sect. 2, novices and experts differ in their problem solving strategies. While novices have to rely on general problem solving strategies like means-ends analysis, experts can rely on plan schemata. Since process modelers in practical settings are often not expert modelers, but rather casual modelers with a basic amount of training [18], we do not require modeling experts for our study. To avoid, however, that difficulties are caused by unfamiliarity with the tool, rather than by difficulties with the tasks themselves, we require subjects to be moderately familiar with process modeling as well as change patterns. For this, subjects are trained using theoretical backgrounds of change patterns, but also obtain hands-on experience in the creation of process models using change patterns to guarantee that they are sufficiently literate in change pattern usage. Regarding process modeling experience and experience in change patterns usage we assume a relatively homogeneous group, which is tested ex-post. Choosing subjects moderately familiar with process modeling and change patterns usage allows us to make statements about casual modelers that cannot be generalized to modeling experts.

Independent Variable and Factor Levels. As independent variable we consider the *nesting depth* of the solution model with factor levels: *high* and *low*.

Objects. As outlined in Sect. 1, the creation of a process model requires the process modelers to construct a mental model (i.e., *internal representation*) of the requirements to be captured in the process model [4] and to map this mental model to the constructs provided by the modeling language (i.e., creating an *external representation* of the domain [4]). Tasks should be designed such that it can be ensured that observed difficulties are caused by change patterns usage rather than problems understanding the domain and constructing the mental model (which would also exist when using change primitives for model creation). Therefore, to factor domain influences out, participating subjects are asked to re-model a process (denoted as reference model in the following) starting from an empty modeling canvas. For this, process designers have to apply a sequence of change patterns to incrementally re-construct the given reference model starting from the empty model canvas. In addition, activities are labeled A, B, C to reduce the potential impact of domain knowledge.

Since the experiment aims to compare the cognitive complexity of using change patterns depending on the nesting depth, two versions (with high and low nesting dept) of the modeling task have to be designed, making sure that both tasks differ only in their nesting depth and not in other model characteristics. From research into process model quality we know, for example, that the size of the process model impacts model comprehension [19] and that different control-flow constructs do not have the same cognitive complexity [20]. As a consequence, these factors have to be controlled when designing the material for the experiment. Our intention therefore is to choose 2 models with the same number of activities, the same change patterns, and the same minimum number of change patterns needed to construct the solution.

Response Variables. To operationalize cognitive complexity of change patterns usage we consider, (1) accuracy, i.e., how close the subject's solution is to the reference model, (2) efficiency, i.e., how many detours it takes them to reach the solution, (3) speed, i.e., how fast they create the solution, and (4) the required mental effort. To measure accuracy we consider *product deviations*, i.e., discrepancies between the modeler's solution and the reference model. For example, a process model which contains two product deviations, requires the application of two change patterns to transform that model into the reference model. To operationalize efficiency, we consider *process deviations* measuring the modeler's detours until coming up with the solution. Process deviations are calculated as difference between the number of applied change patterns to reach the solution and the minimum number of change patterns required for the task. Finally, we consider the *time* needed to accomplish the task (i.e., speed) as well as the required *mental effort*, measured using a questionnaire after the task [21].

Hypotheses. This leads us to the following null hypotheses.

- $H_{1,0}$: High nesting depth does not lead to significantly more product deviations when compared to low nesting depth.
- $H_{2,0}$: High nesting depth does not lead to significantly more process deviations when compared to low nesting depth.
- $H_{3,0}$: High nesting depth does not require significantly more time when compared to low nesting depth.
- $H_{4,0}$: High nesting depth does not impose a significantly higher mental effort when compared to low nesting depth.

Instrumentation. For data collection Cheetah Experimental Platform [22] is used, logging given answers (e.g., demographic data), the time to accomplish the tasks, and all model interactions to obtain process deviations.

Experimental Design. The experiment is conducted as balanced single factor experiment with repeated measurements. Prior to the experiment a familiarization phase takes place (i.e., subjects are trained using change patterns). Subjects are then randomly assigned to two groups of equal size, subsequently referred to as G1 and G2. To provide a balanced experiment with repeated measurements, the overall procedure consists of two runs. In the first run G1 applies factor level *low nesting depth*, G2 factor level *high nesting depth*. In the second run, factor levels are switched and G1 applies factor level *high nesting depth*, G2 factor level *low nesting depth* to the same object. Choosing such a cross design is an additional measure to counter potential learning effects.

4 Summary

While the process of creating process models using change primitives has caused some interest in recent years [4–7], our understanding of the process of creating process models using change patterns is limited. This paper proposes a research design to investigate the PPM using change patterns in more detail. In particular, the impact of nesting depth on the cognitive complexity of creating models is examined. Results of the experiment will provide a better understanding of the PPM using change patterns and help to understand how to design tool-support for change patterns based modeling.

References

1. Kock, N., Verville, J., Danesh-Pajou, A., DeLuca, D.: Communication flow orientation in business process modeling and its effect on redesign success: Results from a field study. DSS 46, 562–575 (2009)
2. Mendling, J., Verbeek, H.M.W., van Dongen, B.F., van der Aalst, W.M.P., Neumann, G.: Detection and prediction of errors in EPCs of the SAP reference model. DKE 64, 312–329 (2008)
3. Weber, B., Reichert, M., Rinderle, S.: Change Patterns and Change Support Features - Enhancing Flexibility in Process-Aware Information Systems. DKE 66, 438–466 (2008)
4. Soffer, P., Kaner, M., Wand, Y.: Towards Understanding the Process of Process Modeling: Theoretical and Empirical Considerations. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM 2011 Workshops, Part I. LNBIP, vol. 99, pp. 357–369. Springer, Heidelberg (2012)
5. Pinggera, J., Zugal, S., Weidlich, M., Fahland, D., Weber, B., Mendling, J., Reijers, H.A.: Tracing the Process of Process Modeling with Modeling Phase Diagrams. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM 2011 Workshops, Part I. LNBIP, vol. 99, pp. 370–382. Springer, Heidelberg (2012)
6. Claes, J., Vanderfeesten, I., Reijers, H.A., Pinggera, J., Weidlich, M., Zugal, S., Fahland, D., Weber, B., Mendling, J., Poels, G.: Tying Process Model Quality to the Modeling Process: The Impact of Structuring, Movement, and Speed. In: Barros, A., Gal, A., Kindler, E. (eds.) BPM 2012. LNCS, vol. 7481, pp. 33–48. Springer, Heidelberg (2012)
7. Pinggera, J., Soffer, P., Zugal, S., Weber, B., Weidlich, M., Fahland, D., Reijers, H.A., Mendling, J.: Modeling Styles in Business Process Modeling. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Wrycza, S. (eds.) EMMSAD 2012 and BPMDS 2012. LNBIP, vol. 113, pp. 151–166. Springer, Heidelberg (2012)
8. Weber, B., Pinggera, J., Torres Bosch, V., Reichert, M.: Change patterns in use: A critical evaluation. In: Proc. BPMDS (accepted, 2013)
9. Gray, P.: Psychology. Worth Publishers (2007)
10. Tracz, W.: Computer programming and the human thought process. Software: Practice and Experience 9, 127–137 (1979)
11. Jeffries, R., Turner, A., Polson, P., Atwood, M.: The Process Involved in Designing Software. In: Cognitive Skills and Their Acquisition, pp. 255–283. Erlbaum (1981)
12. Rist, R.: Schema Creation in Programming. Cognitive Science 13, 389–414 (1989)

13. Kant, E., Newell, A.: Problem Solving Techniques for the design of algorithms. Information Processing & Management 20, 97–118 (1984)
14. Anderson, J.: Acquisition of cognitive skill. Psychological Review 89, 369–406 (1982)
15. Guindon, R., Curtis, B.: Control of cognitive processes during software design: what tools are needed? In: Proc. CHI 1988, pp. 263–268 (1988)
16. Sweller, J.: Cognitive load during problem solving: Effects on learning. Cognitive Science, 257–285 (1988)
17. Brooks, R.: Towards a theory of the cognitive processes in computer programming. International Journal of Man-Machine Studies 9, 737–751 (1977)
18. Pinggera, J., Zugal, S., Weber, B., Fahland, D., Weidlich, M., Mendling, J., Reijers, H.A.: How the Structuring of Domain Knowledge Can Help Casual Process Modelers. In: Parsons, J., Saeki, M., Shoval, P., Woo, C., Wand, Y. (eds.) ER 2010. LNCS, vol. 6412, pp. 445–451. Springer, Heidelberg (2010)
19. Mendling, J., Strembeck, M., Recker, J.: Factors of process model comprehension - findings from a series of experiments. DSS 53, 195–206 (2012)
20. Figl, K., Laue, R.: Cognitive complexity in business process modeling. In: Mouratidis, H., Rolland, C. (eds.) CAiSE 2011. LNCS, vol. 6741, pp. 452–466. Springer, Heidelberg (2011)
21. Zugal, S., Pinggera, J., Reijers, H., Reichert, M., Weber, B.: Making the Case for Measuring Mental Effort. In: Proc. EESSMod 2012, pp. 37–42 (2012)
22. Pinggera, J., Zugal, S., Weber, B.: Investigating the Process of Process Modeling with Cheetah Experimental Platform. In: Proc. ER-POIS 2010, pp. 13–18 (2010)

Enabling Personalized Process Schedules with Time-Aware Process Views

Andreas Lanz, Jens Kolb, and Manfred Reichert

Institute of Databases and Information Systems, Ulm University, Germany
{andreas.lanz,jens.kolb,manfred.reichert}@uni-ulm.de

Summary. Companies increasingly adopt process-aware information systems (PAISs) to model, enact, monitor, and evolve their business processes. Although the proper handling of temporal constraints (e.g., deadlines and minimum time lags between activities) is crucial for many application domains, existing PAISs vary significantly regarding the support of the temporal perspective of a business process. In previous work, we introduced characteristic time patterns for specifying the temporal perspective of PAISs. However, time-aware process schemas might be complex and hard to understand for end-users. To enable their proper visualization, therefore, this paper introduces an approach for transforming time-aware process schemas into enhanced Gantt charts. Based on this, a method for creating personalized process schedules using process views is suggested. Overall, the presented approach enables users to easily understand and monitor time-aware processes in PAISs.

Keywords: Human-centric PAIS, Temporal Perspective, Time Patterns.

1 Introduction

Companies strive for improved life cycle support of their business processes [1]. In particular, IT support for modeling, enacting, and analyzing these processes results in competitive advantages [2,3]. In this context, process-aware information systems (PAISs) offer promising perspectives for process automation. In particular, they allow defining a business process in terms of an explicit *process schema* and executing respective *process instances* in a controlled and efficient manner [1].

Existing PAISs vary significantly regarding their support of the temporal perspective of processes [4,5]. However, integrating the temporal perspective in a PAIS is crucial, since most business processes must obey temporal constraints [6]. Moreover, in many application domains the proper handling of temporal constraints is vital in order to successfully complete a process (e.g., flight planning, patient treatment, and automotive engineering) [7,8]. By contrast, different kinds of planning tools (e.g., project management tools) exist, providing sophisticated facilities for handling and visualizing time constraints. However, these tools lack an operational process support, which is needed for proper visualization and systematic analysis of the temporal perspective in PAISs.

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 205–216, 2013.

Generally, when incorporating the temporal perspective complex and large process schemas may result, which are difficult to comprehend for users. In particular, it is important that all users involved in a business process are aware of respective temporal constraints [9]. To tackle this challenge, we introduce an approach transforming a time-aware process schema to an extended version of Gantt charts. Note that Gantt charts are well known in project management [10]. In particular, Gantt charts allow users to easily perceive and assess temporal properties of time-aware processes. However, despite their simplicity, representing an entire time-aware process schema as Gantt chart might be complex and inappropriate for end-users. To address this issue, we provide mechanisms for abstracting a process schema (and the respective Gantt chart) to meet specific requirements of a particular user. This allows providing personalized process schedules for each user and thus reducing the complexity of time-aware process schemas.

Sect. 2 discusses related work. Sect. 3 introduces fundamentals required to understand this paper. Sect. 4 describes the transformation of time-aware processes to Gantt charts. Sect. 5 shows how personalized process schedules can be created based on respective Gantt charts. Finally, Sect. 6 concludes the paper.

2 Related Work

Gantt charts have been used for visualizing project plans for a long time [11,10]. Nowadays, almost all project management tools offer a Gantt chart-based appearance of project plans [10]. Furthermore, extensions to Gantt charts have been proposed in different application domains. Such an approach is provided by AsbruView [12], which was originally introduced for therapy planing; in particular, it extends Gantt charts with minimum/maximum durations and provides basic support for loops as well as conditional routing. Still, it does not consider some of the specific requirements found in the context of PAISs (e.g., time lags between activities).

In turn, considerable research has been conducted concerning the modeling and verification of time-aware processes [13,5,14,15]. However, respective approaches either rely on traditional process appearances (e.g., BPMN) for visualizing time-aware processes or do not consider visualization issues at all. Closest to this work is the proposal made by Eder et al. regarding personal schedules [9]. In particular, a personal schedule contains all (future) activities assigned to a specific user and suggests an optimal execution order for them according some optimization strategy. Thereby, the activities of all process instances executed in the PAIS are considered, i.e., the activities of a personal schedule may originate from instances running on different process schemas.

3 Backgrounds

This section provides basic notions. Sect. 3.1 discusses fundamentals on time-aware processes, while Sect. 3.2 introduces Gantt charts.

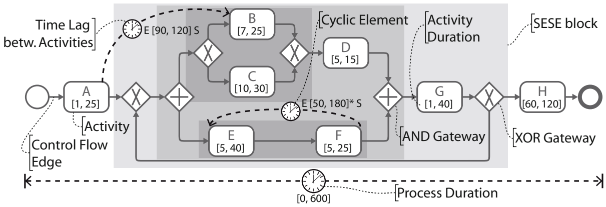


Fig. 1. Well-structured Time-aware Process Schema

3.1 Fundamentals on Time-Aware Process Schema

A *process schema* is described in terms of a directed graph whose node set comprises *activities* and *gateways*. Thereby, an activity corresponds to a human task, requiring user interaction, or an automated task. In turn, gateways may be categorized as *AND*/*XOR*-gateways and may be used for modeling parallel/conditional branchings, and loops. (Note that the latter are expressed through XOR-gateways.) Control edges between activities and/or gateways represent precedence relations, i.e., the *control flow* of the process schema (cf. Fig. 1).

This paper, presumes that process schemas are *well-structured* [16], i.e., sequences, branchings (of different semantics), and loops are specified as blocks with well-defined start and end nodes having the same gateway type. These blocks, also known as *SESE (single-entry-single-exit)* blocks (cf. Fig. 1), may be nested, but are not allowed to overlap.

Regarding the temporal perspective, the following time patterns (TP) are considered in the following: *time lags between two activities* (TP1), *duration* (TP2) of activities and processes, and *cyclic elements* (TP9). These pattern are selected since they are the most relevant ones in practice. A full list of time patterns for PAISs as well as their formal semantics are described in [4,5,17].

An **activity duration** (TP2) restricts the time span allowed for executing an activity, i.e., the time span between the start and end events of the activity [5,17]. We assume that each activity in a process schema has an assigned duration. Activity durations are expressed in terms of a time interval $[MinD, MaxD]$ with $1 \leq MinD \leq MaxD$ (cf. Fig. 1). In addition, a process schema itself may have a **process duration** (TP2) representing the allowed time span between the start and end of corresponding process instances [5,17].

Time lags between two activities (TP1) restrict the time span between start/end events of two activities [5,17]. Such a time lag may not only be defined between directly succeeding activities, but between arbitrary activities, presuming they may be conjointly executed in the context of the same process instance. A time lag is visualized by a dashed edge with a clock (cf. Fig. 1). The label of the edge specifies the constraint: $\langle I_S \rangle \, [MinD, MaxD] \, \langle I_T \rangle$, where $\langle I_S \rangle \in \{S, E\}$ and $\langle I_T \rangle \in \{S, E\}$ mark the events (i.e., start/end) of the source/target activity; e.g., $S[10, 20]E$ expresses that the time lag between the start of the source activity and the end of target activity shall be between 10 and 20 time units.

Cyclic elements (TP9) restrict the time span between activity instances of different iterations of a loop [5,17]. This includes instances of the same activity as well as different activities of a loop. Like time lags, a cyclic element is visualized by a dashed edge with a clock between the source and target activity (cf. Fig. 1). To distinguish between the two, the label of a cyclic element is annotated with a "*" next to the allowed range: $\langle I_S \rangle\, [MinD, MaxD]^* \,\langle I_T \rangle$.

3.2 Fundamentals on Gantt Charts

Gantt charts are used to visualize schedules in the context of project management [11,10]. Generally, a Gantt chart is a bar chart that displays *activities* of a project on a time line together with their temporal relationships. Each activity has a dedicated start and end time. For example, in Fig. 2, activity A is planned to start at $t = 0$ and end at $t = 15$. The horizontal length of an activity represents its duration. Usually the average duration of activities is visualized. In particular, it is not possible to visualize the minimum and maximum duration of an activity. Directed edges between activities represent temporal relationships. To be more precise, Gantt charts support *start-start*, *end-start*, and *end-end* relationships. A *start-start* relationship expresses that both activities start at same time (e.g., B and C in Fig. 2). In turn, an *end-start* relationship indicates that an activity may start after finishing another one (e.g., A and B). Finally, activities with *end-end* relationship must be finished at same time (e.g., C and D).

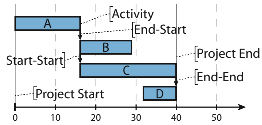


Fig. 2. Gantt Chart

4 Gantt-Based Visualization of Time-Aware Processes

4.1 Transforming Process Schemas to Gantt Charts

This section shows how to transform a well-structured time-aware process to a corresponding Gantt chart. Compared to traditional Gantt charts, however, time-aware process schemas are more expressive (cf. Sect. 3), e.g., considering minimum and maximum activity durations, minimum and maximum time lags between activities, exclusive choices, and loops. Consequently, Gantt charts must be extended to allow for a mapping of time-aware process schemas to them. We denote this extension as *extended Gantt* (eGantt) chart.

Activity Transformation. When transforming a time-aware process schema to an eGantt chart, each process activity A must be mapped to a corresponding eGantt activity A (cf. Fig. 3). Thereby, the length of the bar is chosen according to the minimum duration $MinD$ of A. In addition, a dashed bar is appended visualizing the maximum duration $MaxD$ of A, i.e., length of the dashed bar is $MaxD{-}MinD$. In case $MaxD{=}MinD$, the dashed bar is omitted. Overall, this allows users to easily perceive and assess temporal properties of activities.

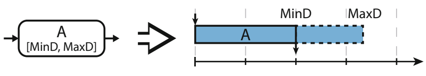


Fig. 3. Visualizing an Activity

Sequence Block Transformation. To transform a sequence of process activities to an eGantt chart, the temporal relationships between the activities must

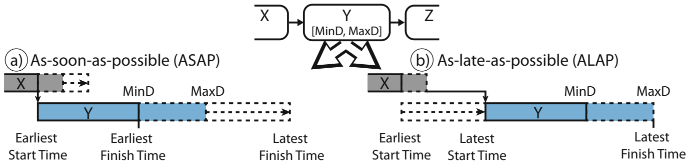


Fig. 4. Time-centric Process Appearance Alternatives

be taken into account. However, since actual activity duration is unknown at build-time no distinct position exists to place the start of succeeding activities on the time line. Even if the minimum/maximum duration of each activity is known at build-time the actual duration will be only known at run-time after completing the activity. To address this, for each process activity, four time values are calculated based on the temporal information provided by the process schema [13,14]: *earliest start time* (EST), *earliest finish time* (EFT), *latest start time* (LST), and *latest finish time* (LFT). Note that these values are known from project planning techniques as well [18]. In particular, they provide a reference frame for the temporal properties of a process schema. Moreover, these values enable us to derive the *critical path* of a process schema which is essential for evaluating its temporal properties. Based on these values, eGantt defines two time-centric process appearances:

- *As-Soon-As-Possible* (ASAP): Each activity is assumed to be started at its earliest possible start time and to be completed after its minimum duration (i.e., at its EFT). Consequently, the succeeding eGantt activity (i.e., the respective bar) starts at its EST (cf. Fig. 4a).
- *As-Late-As-Possible* (ALAP): Each activity is assumed to be started at its latest possible start time and to be completed at its LFT. Hence, the succeeding eGantt activity starts at its LST (cf. Fig. 4b).

To be able to fully assess temporal properties of a process schema, in general, it is necessary to know both the EST and LFT of the corresponding activities. Hence, when choosing ASAP as eGantt appearance, a dashed bar with an arrow is attached to each eGantt activity visualizing its LFT (cf. Fig. 4a). In turn, when choosing ALAP as eGantt appearance, a dashed bar with arrow is placed before the respective eGantt activity visualizing its EST (cf. Fig. 4b).

eGantt activities are connected by arrows indicating their temporal relationship (i.e., precedence relations). Regarding ASAP appearance, the arrow starts at the EFT of the first and ends at EST of the second activity (cf. Fig. 4a). Concerning ALAP appearance, the arrow starts at LFT of the first and ends at LST of the second activity (cf. Fig. 4b). Due to lack of space, we focus on the ASAP appearance for the remainder of this paper.

Following this, time lags between activities need to be considered. For each time lag an arrow is added between the respective activities. Depending on the type of eGantt appearance (i.e., ASAP or ALAP), the arrow connects the EST/EFT or LST/LFT of the activities according to the kind of time lag (i.e., start-start, start-end, end-start, or end-end). Note that, time lags affecting the

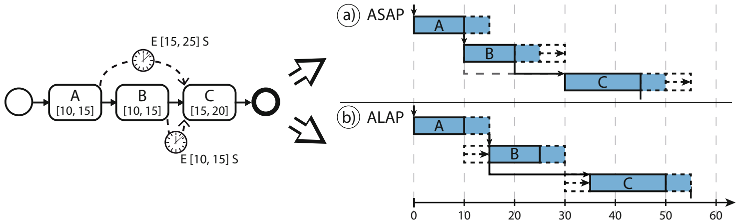


Fig. 5. Example of an Activity Sequence Visualization

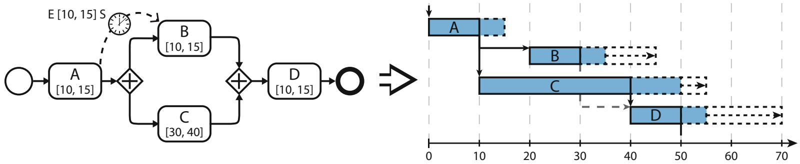


Fig. 6. Parallel Gateway Visualization (ASAP)

start or end time of an activity are taken into account through calculating respective time values (i.e., EST, LST, EFT, and LFT).

Example 1 (Activity Sequence). Consider the process schema depicted in Fig. 5. It contains three sequential activities with defined time lags between A and C as well as B and C. In particular, the time lag between B and C requires that the earliest start time of C is 10 time units after completion of B. Consequently, the EST of C must be delayed by 10 time units (cf. Figs. 5a+b). In turn, the time lag between A and C requires that C must not be started more than 25 time units after completing A. Note that this has no impact on the EST, but requires the LST of activity C to be restricted to time point 35. Regarding Fig. 5a, it is noteworthy that the time lag between A and C has no impact on the earliest start or end time of C. Particularly, the time lag is not part of any critical path.

Generally, temporal relations not part of a critical path are visualized using a dashed grey line instead of a solid one (cf. Fig. 5a). This way, the critical path, being essential for evaluating temporal properties, can be recognized by users.

Parallel Block Transformation. Regarding the transformation of parallel blocks a similar approach is taken. When using ASAP appearance, activities directly succeeding an AND-split (i.e., all branches) may be started after completing the preceding activity. However, it might become necessary to delay the start of one or more branches due to the presence of a time lag (cf. Fig. 6). Following this, the earliest end time of a parallel block is determined by the latest EFT of all branches (cf. Fig. 6). In fact, a subsequent activity must not be started before having completed all branches of the parallel block.

Conditional Block Transformation. Exclusive choices are not supported by traditional Gantt charts. As particular challenge, during run-time, only one of the branches is executed. Generally, it is therefore not possible to determine an exact EST/LST for activities succeeding a conditional block. To support exclusive choices, our eGantt charts introduce distinct *conditional containers*. These

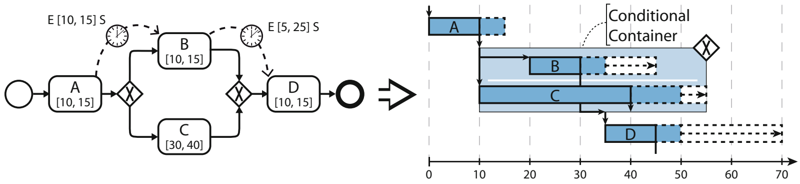


Fig. 7. Conditional Block Visualization (ASAP)

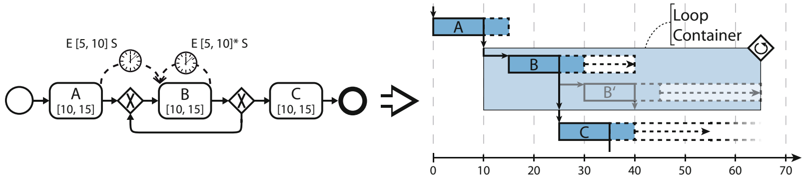


Fig. 8. Loop Block Visualization (ASAP)

encapsulate all branches of an exclusive choice and are marked with a diamond containing an 'X'-symbol on the top right (cf. Fig. 7). Furthermore, respective branches are visually separated through a horizontal line. In the context of ASAP appearance, the activity succeeding the XOR-join is then positioned according to its earliest EST in all alternative execution paths, i.e., branches (cf. Fig. 7).

Loop Block Transformation. Like conditional blocks, loop blocks have not been considered in Gantt charts so far. Therefore, eGantt charts introduce *loop containers* for visualizing loops. In turn, these containers are marked by a diamond containing a loop symbol on the top right (cf. Fig. 7). Generally, loops may be considered as extension of an exclusive choice (see [15]): After each iteration, a decision is made whether to exit the loop or insert another copy of its loop block; eGantt charts adopt this. In particular, when visualizing a loop, a single iteration is added to the loop container in the eGantt chart. Next, a "virtual" second iteration (partially translucent) is added to indicate that the loop body may be executed again (cf. Fig. 8). The activity succeeding the loop is positioned according to the first iteration. Otherwise, it might not be possible to calculate the LST of this activity (i.e., no maximum number of iterations is available). However, to indicate the open end of this eGantt activity, the bar representing the LFT does not end, but fades out after the point of LFT calculated based on the first iteration (cf. Fig. 8). Finally, any subsequent activity is positioned according to the EST/EFT which has been used to position the eGantt activity.

4.2 Executing eGantt-Based Process Appearances

When executing a time-aware process schema respective temporal constraints need to be monitored. Furthermore, users need to be informed about the progress of the process instance and its temporal properties. eGantt supports this by providing a run-time visualization (cf. Fig. 9).

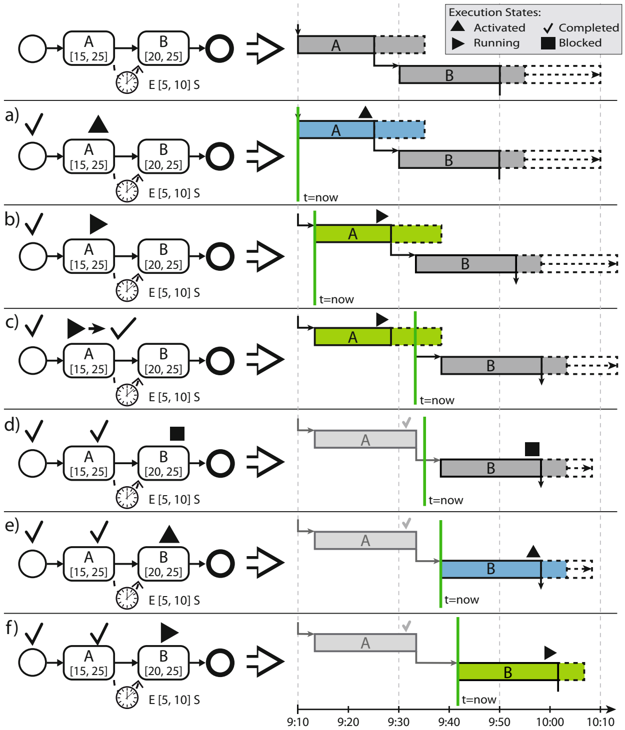


Fig. 9. Run-time Visualization of eGantt Charts

When instantiating a process schema, first of all, the corresponding eGantt chart is configured according to the current time; i.e., the time line is adapted to reflect the time the process is executed (e.g., 9:10 am in Fig. 9). Next, a *progress line* is added to represent the current point in time on the time line (cf. Fig. 9a). Finally, execution state symbols are used to visualize the state of the activities, e.g., an activity may be *activated* (but not yet *running*), *running*, *blocked* (due to a minimum time lag), or *completed*.

When executing a process instance, initially, its first activity enters execution state *activated* (cf. Fig. 9a). To reflect the current time, an activity in this state–together with the progress line–moves along the time line until it starts. Further, all other activities also move along the time line reflecting the pending start of the first activity (cf. Fig. 9b). This visualization allows users to correctly assess future execution times of subsequent activities as well as temporal properties of the process instance at any time.

As soon as the activity is started by a user, it stops moving and switches to execution state *running* (cf. Fig. 9b). At this point, the LFT is the same as the current time plus the maximum duration of the activity, i.e., the bar visualizing the LFT is removed. As soon as the activity duration exceeds the minimum duration, succeeding activities start moving again to reflect this.

When *completing* an activity, the length of its bar is adapted according to its actual execution duration. The positions of subsequent activities on the time line

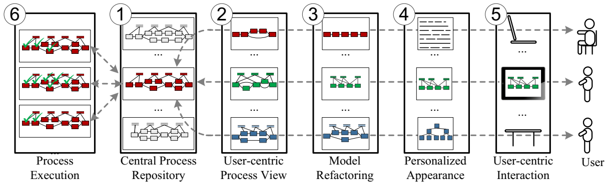


Fig. 10. proView Framework

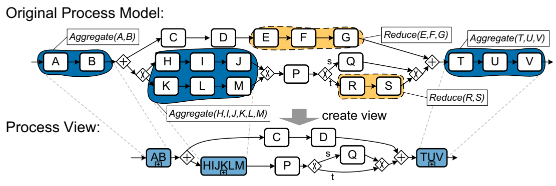


Fig. 11. Example of a Process View

are the adapted according to the actual duration of the preceding activity (cf. Fig. 9c). Next, the succeeding activity is either marked as *blocked* or *activated*. An activity is blocked, if a minimum time lag has not been reached yet (cf. Fig. 9d) and the user must wait before executing it. In turn, an activity in state activated may be started by the user (cf. Fig. 9e). When reaching the end of the process instance, the eGantt chart visualizes the actual durations and execution history of the process instance.

5 Embedding eGantt Charts in a Process Management Framework

We incorporated the presented eGantt charts in the *proView* framework to create personalized process schedules; *proView* aims at supporting users in interacting with large business process schemas and evolving them at a high level of abstraction [19,20]. For this purpose, personalized and updatable process views (cf. Fig. 10, 2) are created for each user role, abstracting from large and complex process schemas stored in the central process repository (cf. Fig. 10, 1).

A process view abstracts from a large process schema by hiding non-relevant process elements (i.e., reduction operation) or by combining them to an abstracted node (i.e., aggregation operation). Fig. 11 gives an example of how such a process view is constructed in *proView* (see [21,22] for details).

When applying view creation operations, superfluous gateways or AND branches without corresponding activities might result. Therefore, the schema of a process view is simplified through behavior-preserving refactorings (cf. Fig. 10, 3), e.g., AND-gateways of a parallel branching with only one remaining branch are removed. Furthermore, *proView* allows transforming the resulting process view into a personalized visual appearance (cf. Fig. 10, 4), e.g., a textual, form-based, and tree-based representation, or—as suggested in this paper—an eGantt chart. Using the resulting visual appearance (cf. Fig. 10, 5) , the user

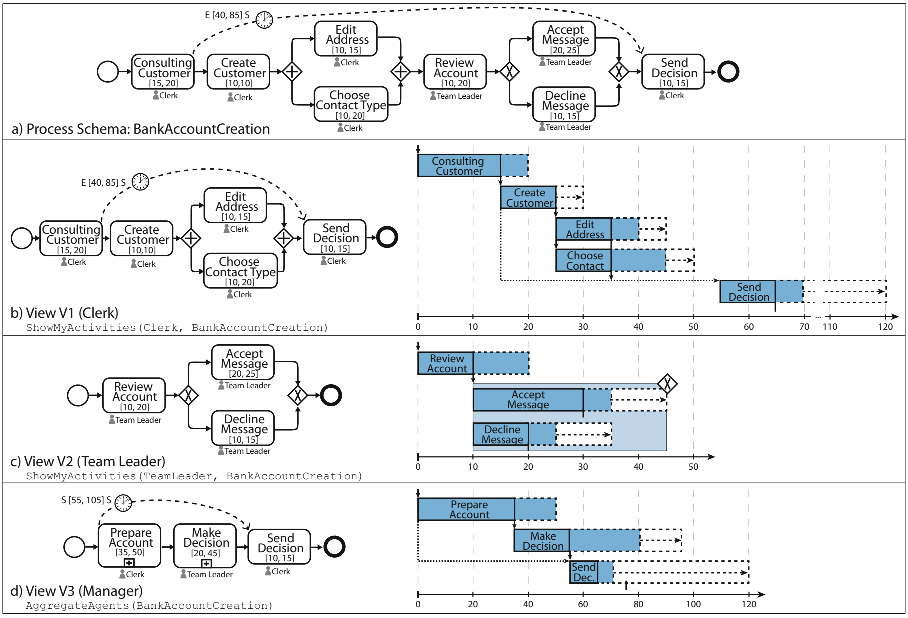


Fig. 12. Creating Personalized Process Schedules

may then read or update the process schema [23,24]. Finally, *proView* supports process execution (cf. Fig. 10, 6). In this context, run-time information is added to process views and process appearances (e.g., eGantt charts).

5.1 Creating Personalized Process Schedules

We present a two-step-method for creating a personalized process schedule visualizing a time-aware process schema for a particular user. To illustrate this method, Fig. 12 introduces a simple scenario related to a *bank account creation* process. This process involves a clerk, team leader, and manager. The clerk consults the customer and prepares the creation of the account. In turn, the team leader checks whether the account creation is accepted or declined. Finally, the clerk informs the customer about the decision. Furthermore, a manager supervises overall process execution. All activities have minimum and maximum durations indicating the time a user is expected to work on the particular activity. Further, there is a time lag expressing that *after* consultation of the customer, he gets the decision after 40 time units the earliest and 85 time units the latest. To create personalized process schedules, for each user as well as for the manager requiring an abstract overview of the process, the following two steps are performed:

Step 1. First, *role-specific process views* are created for the given process schema to abstract from particular process aspects. In our context, three process views are required. High-level view creation operation `ShowMyActivities` creates a process view for roles *clerk* (cf. Fig. 12b) and *team leader* (cf. Fig. 12c) eliminating activities not performed by them. Furthermore, operation `AggregateAgents` combines SESE blocks performed by the same user to one abstracted activity (cf. Fig. 12d). Note that the time lag between activities *Consulting Customer* and *Send Decision* is maintained for process views *V1* and *V3*. In particular, view operation `AggregateAgents` needs to adjust the time lag since the original source activity is aggregated. For this, the source of the time lag is adapted to the start of the aggregated activity *Prepare Account*. View V2 does not include the time lag anymore since it is not relevant for the respective user.

Step 2. The *eGantt chart transformation* is performed by applying the transformations described in Sect. 4 to the created view schemas. Note that the gap before activity *Send Decision* in view V1 is due to combination of the time lag and the elimination of the team leader's activities; i.e., before executing this activity, the clerk must wait until preceding activities of the team leader are completed and the minimum time lag is reached. Since view V2 has no such temporal relationships, no gap is visualized in the eGantt chart. Finally, the eGantt chart of V3 shows the aggregated SESE blocks of the users involved. V3 and its eGantt chart appearance give a comprehensive and abstract overview of the temporal information captured in the time-aware process schema *Bank Account Creation*.

6 Summary and Outlook

We introduced personalized process schedules based on time-aware process schemas. For this, time-aware process schemas are abstracted using high-level view-creation operations. Based on personalized process views, the transformation to extended Gantt charts is described. In particular, our process visualization approach enables users to easily understand the temporal perspective of process schemas. In future work, we evaluate proposed eGantt charts and fully implement them in the *proView* framework including its execution environment. Further, we will develop time-centric view operations as well (e.g., to only show activities executed within a certain time frame). Finally, temporal information shall be extracted from process logs using process mining techniques. In turn, this information will be used to refine respective eGantt charts.

Acknowledgement. The authors would like to acknowledge and thank Martin Sommer for providing first results on this topic.

References

1. Reichert, M., Weber, B.: Enabling Flexibility in Process-aware Information Systems - Challenges, Methods, Technologies. Springer (2012)
2. Lenz, R., Reichert, M.: IT support for healthcare processes - premises, challenges, perspectives. Data & Knowledge Engineering 61(1), 39–58 (2007)
3. Müller, D., Herbst, J., Hammori, M., Reichert, M.: IT support for release management processes in the automotive industry. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 368–377. Springer, Heidelberg (2006)

4. Lanz, A., Weber, B., Reichert, M.: Workflow time patterns for process-aware information systems. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Ukor, R. (eds.) BPMDS 2010 and EMMSAD 2010. LNBIP, vol. 50, pp. 94–107. Springer, Heidelberg (2010)
5. Lanz, A., Weber, B., Reichert, M.: Time patterns for process-aware information systems. Requirements Engineering (2013)
6. Combi, C., Gozzi, M., Juarez, J.M., Oliboni, B., Pozzi, G.: Conceptual modeling of temporal clinical workflows. In: Proc. Int'l Symp. TIME 2007, pp. 70–81. IEEE (2007)
7. Eder, J., Panagos, E., Rabinovich, M.: Time constraints in workflow systems. In: Jarke, M., Oberweis, A. (eds.) CAiSE 1999. LNCS, vol. 1626, pp. 286–300. Springer, Heidelberg (1999)
8. Dadam, P., Reichert, M., Kuhn, K.: Clinical workflows - the killer application for process-oriented information systems. In: Proc. Int'l Conf. BIS 2000, pp. 36–59 (2000)
9. Eder, J., Pichler, H., Gruber, W., Ninaus, M.: Personal schedules for workflow systems. In: van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 216–231. Springer, Heidelberg (2003)
10. Maylor, H.: Beyond the gantt chart: Project management moving on. European Management Journal 19(1), 92–100 (2001)
11. Clark, W.: The Gantt Chart. The Ronald Press Co. (1922)
12. Kosara, R., Miksch, S.: Metaphors of movement: a visualization and user interface for time-oriented, skeletal plans. Artif. Intell. Med. 22(2), 111–131 (2001)
13. Bettini, C., Wang, X.S., Jajodia, S.: Temporal reasoning in workflow systems. Distributed and Parallel Databases 11(3), 269–306 (2002)
14. Eder, J., Panagos, E.: Managing time in workflow systems. In: Fischer, L. (ed.) Workflow Handbook 2001, pp. 109–132. Future Strategies Inc. (2000)
15. Combi, C., Gozzi, M., Posenato, R., Pozzi, G.: Conceptual modeling of flexible temporal workflows. ACM Trans. Auto. & Adapt. Syst. 7(2), 19:1–19:29 (2012)
16. Kiepuszewski, B., ter Hofstede, A., Bussler, C.: On structured workflow modelling. In: Wangler, B., Bergman, L.D. (eds.) CAiSE 2000. LNCS, vol. 1789, pp. 431–445. Springer, Heidelberg (2000)
17. Lanz, A., Reichert, M., Weber, B.: A formal semantics of time patterns for process-aware information systems. TR UIB-2013-02, University of Ulm (2013)
18. Lockyer, K.G., Gordon, J.: Project Management and Project Network Techniques. Prentice Hall (2005)
19. Kolb, J., Kammerer, K., Reichert, M.: Updatable process views for user-centered adaption of large process models. In: Liu, C., Ludwig, H., Toumani, F., Yu, Q. (eds.) ICSOC 2012. LNCS, vol. 7636, pp. 484–498. Springer, Heidelberg (2012)
20. Kolb, J., Kammerer, K., Reichert, M.: Updatable process views for adapting large process models: The proview demonstrator. In: Proc. BPM 2012 Demo Track (2012)
21. Reichert, M., Kolb, J., Bobrik, R., Bauer, T.: Enabling personalized visualization of large business processes through parameterizable views. In: Proc. Symp. Appl. Comp. (SAC 2012) (2012)
22. Kolb, J., Reichert, M.: Data flow abstractions and adaptations through updatable process views. In: Proc. Symp. Appl. Com. (SAC 2013) (to appear, 2013)
23. Kolb, J., Rudner, B., Reichert, M.: Towards gesture-based process modeling on multi-touch devices. In: Bajec, M., Eder, J. (eds.) CAiSE 2012 Workshops. LNBIP, vol. 112, pp. 280–293. Springer, Heidelberg (2012)
24. Kabicher-Fuchs, S., Rinderle-Ma, S., Recker, J., Indulska, M., Charoy, F., Christiaanse, R., Dunkl, R., Grambow, G., Kolb, J., Leopold, H., Mendling, J.: Human-centric process-aware information systems. CoRR abs/1211.4 (2012)

Automatic Business Process Model Translation with BPMT

Kimon Batoulis[1], Rami-Habib Eid-Sabbagh[1], Henrik Leopold[2], Mathias Weske[1], and Jan Mendling[3]

[1] Hasso-Plattner-Institute, University Potsdam, Germany
kimon.batoulis@student.hpi.uni-potsdam.de,
{rami.eidsabbagh,mathias.weske}@hpi.uni-potsdam.de
[2] Humboldt-Universität zu Berlin, Berlin, Germany
henrik.leopold@wiwi.hu-berlin.de
[3] Wirtschaftsuniversität Wien, Vienna, Austria
jan.mendling@wu.ac.at

Abstract. Nowadays, many enterprises use business process models for documenting and supporting their operations. As many enterprises have branches in several countries and provide similar services throughout the globe, there is high potential for re-using these process models. However, the language barrier is a major obstacle for the successful re-use of process models, especially in multi-national companies. In this paper, we address this problem by presenting the Business Process Model Translator (BPMT), a technique for the automated translation of business process models that eases the re-use of business process models and reduces redundant work in multi-national companies. It builds upon the state-of-the-art machine translation system Moses and extends it with word and translation disambiguation considering the context of the domain. As a result, the BPMT can successfully deal with the compact and special language fragments that are typically found in business process models. A two-fold evaluation with the BLEU metric and an expert survey showed improvements of our approach over Moses.

Keywords: business process models, statistical machine translation, word sense disambiguation, translation disambiguation.

1 Introduction

Business process model collections are important assets of companies. Large enterprises operate with a network of branches in multiple countries with different official languages. Delivering the same services in new locations, the transfer, adaptation, and re-use of their business processes represents a major advantage [1]. However, language barriers impede a straightforward adaptation and consequently the re-use of existing process models.

As an example, consider the following scenario. An employee of a new subsidiary may want to define the new branch's order process as this does not yet exist in the new location. Hence, he will search in the company's global process

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 217–228, 2013.

repository for a relevant business process. This imposes several problems. First, the employee might not be able to find a specific model in the global repository as the processes are modeled in a foreign language and the search in his local language will not deliver any search results. Second, even if the employee speaks other languages and finds an according process model, he will not be able to use it as the majority of his colleagues may only speak the local language. In order to adapt the process model for re-use in the whole branch, a manual translation of all process labels would be required. For a large process collection this results in a massive amount of work which is associated with considerable costs. Accordingly, a technique that facilitates the automated translation of business process models could significantly increase the re-use of process models and hence reduce redundant work efforts as well as costs. As a result, multinational communication and collaboration among employees could be strengthened and eased.

Nevertheless, prior research highlighted the challenges that are associated with the automatic analysis of natural language in process models [2]. Since process model labels do not contain full and grammatically correct sentences, the application of standard tools for natural language processing such as parsers turned out to be hardly possible. Some authors even recommended to avoid the application of natural language processing in process models because of these issues [3]. Accordingly, the straightforward application of machine translation techniques for the translation of process models does not represent a promising strategy.

In this paper, we present the Business Process Model Translator (BPMT), a technique for the automated translation of business process model labels. In order to deal with the specific challenges of the natural language in business process models, we extend the state-of-the-art machine translation system Moses. By introducing word and translation disambiguation our technique includes the context of the domain and hence yields more stable results than a naive application of Moses. We use activity labels as use case for translation as they constitute an important part of business process models. While the presented approach is language-independent, we demonstrate the applicability of the technique by translating process models from English to German.

The remainder of this paper is structured as follows. Section 2 gives a brief introduction to machine translation and the Moses translation system.

Section 3 describes problems concerning process model label translations and presents our Moses extension, the Business Process Model Translator (BPMT). Section 4 presents a two-fold evaluation of our translation approach, followed by related work in Section 5. Section 6 concludes the paper.

2 Background

The general objective of machine translation is the automatic mapping of text from a source language (like English) to a target language (like German). As it is the case for all tasks of natural language processing, this problem can be tackled by using linguistic or statistical information about the languages in question.

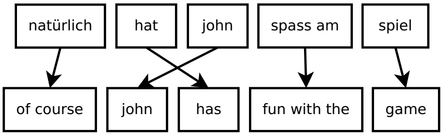


Fig. 1. Phrasal translation from German to English [6, p. 184]

Of course, it is also possible to use a combination of both [4,5]. In this paper, however, we concentrate on statistical methods. These methods are based on the basic model of statistical machine translation consisting of two components, a *translation model* and a *language model* [4]. The former gives information about the probability of the translation of a string from a source to a target language and the latter the general probability of the target language string. The target language string that maximizes the product of those two probabilities is then chosen as the source language's translation.

Our approach, the Business Process Model Translator (BPMT), is based on the open source statistical machine translation toolkit *Moses* [6]. Moses builds on the idea of phrase-based translation, which means that the probabilities of the translation model are not given for individual words or entire sentences but for phrases, a syntactical group consisting of one or more words. Currently, this procedure represents the state-of-the-art in modern machine translation [4]. As an example, consider the translation of the German sentence *natürlich hat john spass am spiel* to English, where the overall translation consists of the combination of translations of individual phrases as illustrated in Fig. 1. Note that it is possible to reorder phrases during translation. Moses makes it possible to extend the basic model of statistical machine translation mentioned above with weighted *feature functions* to improve translation quality. The underlying equation is as follows:

$$\arg\max_T P(T|S) = \arg\max_T \prod_i h_i(T,S)^{\lambda_i} \tag{1}$$

where S is the source language, T the target language, h_i the feature functions (e.g. the translation model), and λ_i the corresponding weights.

The translation system created for the BPMT was trained according to the description of building a "baseline system" on the Moses website[1]. This system's model consists of six overall features [7,8]. Both the phrase translation model and the language model were trained using the *News Commentary* corpus[2] as there is no corpus for the business process management domain.

[1] `http://www.statmt.org/moses/?n=Moses.Baseline`.
[2] `http://www.statmt.org/wmt12/translation-task.html`.

3 Business Process Model Translator

When using a translation such as Moses and applying it for the translation of activity labels in business process models, the quality of the translation will certainly not be satisfying. This has one major reason, namely the fact that the language style used in business process models is not representative of the language style of the corpora on which the translation system was trained. On the one hand, business process models use very specific vocabulary that has to be interpreted in the context of the process model's domain. On the other hand, the language is very compact and uses *recurring patterns* of sentence fragments only. As an example, consider the label *creation of master record for tangible assets*. It contains phrases like *master record* and *tangible assets* whose translations are even hard for humans to obtain without complete knowledge of the business context. Moreover, these phrases *must* be translated as a whole since *master* as an individual word has a very different translation than the phrase *master record*. The trained translation system of the previous section would translate this label to German as *Schaffung von Herrn für Sachanlagen*, where the phrase translations are the following:

- *creation of* → *Schaffung von*
- *master* → *Herrn*
- *record for* → *für*
- *tangible assets* → *Sachanlagen*

The system managed to translate the phrase *tangible assets* as a whole, yielding a good translation of this phrase. But *master* was translated in isolation. *Herrn* (Engl. *Mister*) might be a good translation of this word in general, but in this context it is not. The system did not have another choice because the phrase *master record* does not occur in the News Commentary corpus. This example also demonstrates that the phrase alignment illustrated in Fig. 1 is not perfect. It sometimes makes questionable alignments like *record for* → *für*. With this alignment, only *for* is translated, ignoring the word *record*.

Unfortunately, the problem of data sparseness is not restricted to uncommon vocabulary. Even the phrase *check availability* neither occurs in the News Commentary nor in Europarl, another freely available language corpus. The only phrase in the Europarl corpus that is similar to *check availability* is *checking the availability*. Consequently, the words have to be translated individually, yielding *Schach Verfügbarkeit* (Engl. *chess availability*) as the *most probable* translation to German. In this case, the ambiguous word *check* was translated to *Schach* (Engl. chess) based on a wrong meaning.

3.1 Moses Translation Candidates

To solve this problem the BPMT makes use of the fact that Moses tries to find the most probable translation from a list of *translation candidates*. Table 1 shows an excerpt of the possible translations of *check availability* sorted by their probability (highest first) according to Moses. This table demonstrates the fact

Table 1. Translation candidates of *check availability* according to Moses

#	Translation Candidate
1	Schach Verfügbarkeit
2	Schach halten Verfügbarkeit
...	...
51	Verfügbarkeit überprüfen
...	...
553	Kreditfähigkeit Einkommenssegmenten
...	...

Table 2. Assignment of categories to words of labels in business process models

Label	Action	Business Object	Addition
Creation of master record for tangible assets	create	master record	for tangible assets
program analysis	analyse	program	

that although phrase translation was not possible there is a correct word translation including word re-ordering (#51) in which *check* was correctly translated to *überprüfen* instead of *Schach* (Engl. *chess*). How can we select #51 instead of #1? Our approach is based on the idea of *word sense disambiguation*, finding the meaning of a word in a given context. For example, we would have to find out that *check* is used as in *check the brakes* (translation *überprüfen*) and not as in *the chess player's king is placed in check* (translation *Schach*). This involves two things: First, the part-of-speech (POS) of *check* (whether it is a noun, verb, adjective etc.) needs to be identified. Second, after having determined that *check* is used as a verb, the actual disambiguation must be done. POS tagging of the words of a label is done on the basis of a label refactoring tool described in [2]. The refactorer assigns the categories *action*, *business object* and *addition* to the words of the activity labels in business process models (see Table 2). Moreover, it converts the labels to the recommended verb-object style [9]. This information is used to infer the words' part-of-speeches: actions are declared as verbs and business objects as nouns. Additions can contain various part-of-speeches, thus it would be unwise to assign a single one to them. Instead, the *Stanford Part-of-Speech Tagger*[3] is used to assign the (potentially) correct POS. The tagger is not used for all words as taggers require *syntactically correct* context to give accurate results, which labels of process models typically do not provide.

Knowing the meaning of a word in a given context, provides valuable information about its correct translation, which is obvious when looking at the exemplary usage examples of *check* above.

[3] `http://nlp.stanford.edu/software/tagger.shtml`.

3.2 Finding the Best Translation

BPMT's algorithm for finding the best translation will be explained using the example label *check availability*. First, *check* and *availability* are disambiguated. This requires possible definitions of these words, which can be obtained from the English lexical database *WordNet*[4]. Basically, WordNet groups words together based on their meanings. Thus, it is possible to ask WordNet for (i) a definition of a word and (ii) other words that are related to it. Those relations include synonymy, hypernymy[5], hyponymy[6] and many more.

To find the correct definition in a given context, the Lesk algorithm [10] is used. For the BPMT it was implemented similar to [11]. The implementation determines the overlap of the WordNet definition from (i) and (ii) with its context. The BPMT defines the context of a label of a business process model as the concatenation of all the labels of the model. This is acceptable in this situation as we are mainly interested in the words themselves and not their syntactic combinations as it is the case for POS taggers. BPMT then scores the overlaps according to $\text{score} = \sum_{\text{overlap}} \text{length}(\text{overlap})^2$. This gives a higher score to longer *consecutive* overlaps. Thus, a definition that has one one-word sequence overlap and one two-word sequence overlap with the context gets a score of 5.

Having obtained the correct definitions of *check* and *availability*, Moses is asked to provide a list of possible translations of each definition. This helps in extracting the correct translation of *check availability* from the list in Table 1. Again, the algorithm for disambiguating words just described is employed. But now its task is to disambiguate *translations*.

Drawing analogies to the different steps of word sense disambiguation makes this point clear: The input to a word sense disambiguation algorithm is a word and its context. WordNet is asked for possible definitions of the word. The output is the correct definition given the context. Analogously, the input to the BPMT translation disambiguation is a label and all the translations of the definitions of the words of this label. Moses is asked for possible translations of the label. The output is the correct translation given the translations of the definitions.

3.3 BPMT Architecture

The BPMT system consists of two core modules: Moses and the word/translation disambiguator. These modules take care of the translation and the selection of the best result. To feed the BPMT with business process model labels it was integrated into PromniCAT, a platform for research on process model collections [12][7]. Among others, the platform provides utility units to extract business process models and their activity labels from business process model repositories like the BPMAI[8]. The architecture of the BPMT is depicted in Fig. 2 as a UML

4 `http://wordnet.princeton.edu/`.

5 A more general word: *vehicle* is a hypernym of *car*.

6 A more specific word: *car* is a hyponym of *vehicle*.

7 `http://code.google.com/p/promnicat/`.

8 `http://bpt.hpi.uni-potsdam.de/BPMAcademicInitiative/WebHome`.

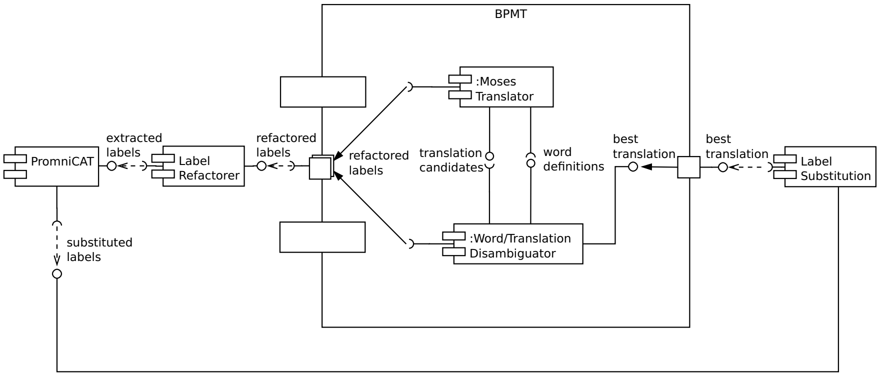


Fig. 2. The component architecture of the BPMT

component diagram. Business processes of the BPMAI modeled in BPMN 2.0 are selected from the research platform's process repository. The activity labels are extracted and passed through the label refactoring tool described in Section 3. The refactored labels are then handed over to the BPMT where Moses interacts with the translation and word sense disambiguation module in order to find the best translation. Finally, the original label is replaced by its translation in the process model and the translated model is stored as a new revision in the research platform's repository.

4 Evaluation

An interesting question concerning the translations of the BPMT is how probable it is that they are different from the translations favored by Moses. In those cases the actual translation is not the first of the list of translation candidates but some other translation in that list. In order to answer this question, we let the two systems translate a subset of the activity label data set from the BPMAI, totaling 2084 labels. The evaluation revealed that nearly 63% of the Moses translations are discarded. Table 3 shows some of those differences. Basically, there are four classes of results: Rows (1), (2), (3,4) and (5).

The first row shows that there are cases in which the BPMT performs better than Moses. In this case, the BPMT disambiguated *receive* correctly and then chose a better translation for it based on the translations of the definition of *receive* and their overlap with the translation candidates of *receive* provided by Moses. The translation *eine der* for *receive* by Moses (which is incomprehensible) is due to a misalignment during training.

Yet, in some cases, as in row 2, translation quality decreases. This is due to the fact that the BPMT wrongly disambiguated *produce* as *bring out for display*— instead of *create or manufacture a man-made product*. Now, the translation lists of both *bring out for display* and *produce dog food* contain translations with the

Table 3. Comparison of Moses and BPMT translations

Label	Moses	BPMT
receive Mail	einer der Mail	Mail erhalten
produce dog food	Nahrungsmittel zu produzieren Hund	Hund Nahrungsmittel bringen
buy book	kaufen Buch	Buch erwerben
send decision	schicken Entscheidung	Entscheidung schicken
create leased asset master record	verleaste schaffen , die Meister Bilanz	einem gepachteten Stück Vermögenswerte Meister Bilanz schaffen

word ***bringen***. For example, in the first case *für die Tag* ***bringen*** and in the second case *Hund Nahrungsmittel* ***bringen***. Thus, the BPMT concluded that *Hund Nahrungsmittel bringen* is the correct translation.

BPMT corrects syntactic structures of phrases. This is represented in another class that consists of rows 3 and 4, where all translations are fairly accurate. In this class the translation of an individual word changes slightly (in row 3, *kaufen* is a synonym of *erwerben*) or the syntactic structure of a phrase was changed. E.g., in German it is wrong if a verb in its infinitive form precedes the noun. The BPMT uses the Stanford Tagger to identify if a verb is the first word of a label. In those cases the verb is moved to the end of the label. Consequently, *erwerben Buch* will be changed to *Buch erwerben*. Also, the BPMT removes unnecessary elements for process model labels like the honorific form and commas, among others. For example, the BPMT removes the honorific form *Sie* from the translation of *send decision* (row 4) to *schicken Sie Entscheidung* and later switches words as the first word of the translated label is a verb. The resulting translation *Entscheidung schicken* is very good.

The last class demonstrates a substantial problem: If the language used in the label is so specific and compact as in row 5, the list of possible translations will not contain any correct result that could be identified by the BPMT. The phrase *asset master record* must be encountered during training, so that it can be translated as a whole. As neither the News Commentary nor the Europarl corpus contain this phrase, both translations in row 5 are fairly bad.

4.1 BLEU Score

The analysis of the different translations of the 2084 labels suggested that the BPMT performs better than Moses. To validate the results we quantitatively evaluated the BPMT and Moses translations using the BLEU Score [13]. It is computed using an algorithm that scores the translation system output by the number of N-gram overlaps with a reference translation that has to be created manually. Three reference translations were created by us for 207 of the 2084 labels as no reference translations of models from process repositories exist.

Table 4. BLEU Scores for BPMT and Moses translations

	1-grams	**2-grams**	**3-grams**
BPMT	38.96	22.95	14.94
Moses	38.63	21.72	13.82

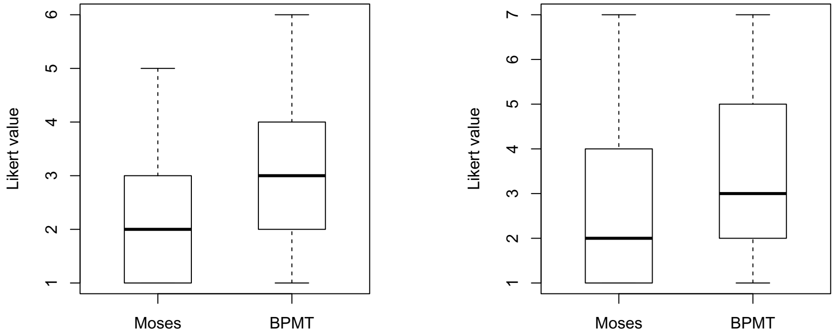
When using the BLEU algorithm one has to decide how long the sequence of overlapping words (i.e. the size of the N-gram) should be. Smaller N-grams yield higher scores, but may also be less meaningful. For example, when using 1-grams, the words that the two translations have in common can occur randomly in the text. In the case of 2-grams they have to occur in common two-word sequences. Since 2-word activity labels are very common in process models, the N-gram scores were computed for $N = 1, 2, 3$. The results are shown in Table 4. As one can see, the two translation systems perform nearly equally well in the choice of their words (1-gram score). Note that this does not imply that they choose the same words. However, larger N-grams lead to a more significant difference between the scores. This means that the BPMT translations are more *fluent* and thus better understandable than the Moses translations.

4.2 Expert Survey

Despite the clear difference between the BLEU scores, this does not represent the actual experienced improvement for two reasons: First, BLEU is rather suited to conduct an evaluation of translations of entire corpora, averaging out errors on individual sentences [13,14]. In contrast, our evaluation data set is comparatively small and only consists of activity labels, not sentences. Second, BLEU does not consider synonymy and other forms of semantic relations—only identical words are scored.

For those reasons we conducted an expert evaluation, in which 20 subjects with a strong BPM background were asked to assess the quality of the translations of both Moses and BPMT. The survey included translations of six randomly selected process models from the BPMAI and was structured as follows: For each process model, the subjects were first asked to rate the translations of the entire models. Afterwards, they rated the translations of the model's labels that differed between Moses and BPMT. The six process models taken together contained 77 labels, 54 (70%) of which were translated differently by Moses and BPMT. The translations were rated using a Likert scale, ranging from 1 (*not acceptable*) to 7 (*excellent*). The ratings are visualized as box plots in Fig. 3a that shows the plots of the ratings of the entire models and Fig. 3b that shows the ratings of the differing labels. The plots illustrate that in both cases the median of the BPMT is one value higher than that of Moses. In addition to the higher BLEU score, this supports the impression that translation quality increases when the BPMT is used to translate process models instead of Moses.

To prove that there is a significant difference between the medians of the translation quality ratings, we conducted the nonparametric sign test. This test



(a) Entire Models (b) Individual Labels

Fig. 3. Box-plots of translation quality ratings for Moses and BPMT

makes no assumptions about the distribution of the population from which the samples are drawn. The results show that for both the ratings of the entire models as well as the individual labels we can reject the hypothesis that there is no difference between the ratings with a confidence of 99.9%.

In summary, an improvement to the overall translation quality has been achieved, but more work has to be devoted to making it more stable so that *every* translation is understandable.

5 Related Work

The work presented in this paper can be related to two major streams of research: statistical machine translation and natural language processing in process models.

Applying natural language processing in process models that pays attention to the structure of process model labels is a quite recent endeavor. In this context, techniques for the enforcement of naming conventions [15] or the refactoring of activity labels have been proposed [2]. Particularly, the latter technique is an important foundation for the technique proposed in this paper. Only by explicitly building on the label components such as action and business object, the translation of the state-of-the-art machine translation system Moses could be improved. Prior works which did not consider the structure of process model labels have been mainly concerned with improving the language in terms of the consistent usage of words [16,17,18]. These works represent complementary techniques to the translation approach as they can help to assure a consistent use of natural language before the model is translated.

Statistical machine translation is nowadays dominated by a phrase-based approach relying on huge bilingual corpora on which the translation engine is trained without using any linguistic information [4]. However, there are several works that try to improve the quality of translations on the basis of semantic information. For example, [19,20] report that they achieve a statistically significantly higher BLEU score by disambiguating the phrases of the source text and thus selecting a translation that matches the obtained definitions. In [21] a system is described that reduces the amount of training data necessary to build a well-performing translator. This is achieved by creating models of the interdependencies of related inflected word forms, which can also help in finding the right word form during translation if not enough context information is available. Also, methods for language-specific sentence-level restructuring transformations are applied after the translation has been obtained. Finally, in [22] a very large amount of context of the target language is used to find and rank N-grams containing the phrase translations of the source text. However, to the best of our knowledge, there are no works that systematically evaluate their techniques using very short phrases only, like those in business process models.

6 Conclusion and Future Work

This paper presented the BPMT, a tool for automatic translation of activity labels of business process models to facilitate their re-use in an international context. Our use case, the translation from English to German, can be easily adapted to different languages by training the BPMT on corpora of the languages in question. The presented concept of word and translation disambiguation does not change. The functionality of the correction of syntactic structure described in Section 4 can be adjusted to the rules of the target language as well. BPMT extends Moses to the specific BPM domain to improve translation quality. Our evaluation results showed that BPMT performs better than Moses by itself.

In future work, two aspects need to be considered: First, language specific tuning of the algorithm could further improve translation quality. For example, moving a verb from the beginning of a label to the end, can lead to problems if the label contains a conjunction of actions. Second, more generally, training of a *factored* translation system [6] should be taken into consideration.

References

1. Indulska, M., Recker, J., Rosemann, M., Green, P.: Business process modeling: Current issues and future challenges. In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) CAiSE 2009. LNCS, vol. 5565, pp. 501–514. Springer, Heidelberg (2009)
2. Leopold, H., Smirnov, S., Mendling, J.: On the refactoring of activity labels in business process models. Inf. Syst. 37(5), 443–459 (2012)
3. Becker, J., Bergener, P., Breuker, D., Räckers, M.: An empirical assessment of the usefulness of weakness patterns in business process redesign. In: 20th European Conference on Information Systems (ECIS 2012), Barcelona, Spain (2012)
4. Jurafsky, D., Martin, J.H.: Speech and Language Processing, 2nd edn. Prentice Hall Series in Artificial Intelligence. Prentice Hall (2008)
5. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach, 2nd edn. Prentice-Hall, Englewood Cliffs (2003)
6. Koehn, P.: MOSES, Statistical Machine Translation System, User Manual and Code Guide (2010)

7. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: Proceedings of the 2003 Conference of the NAACL on Human Language Technology, NAACL 2003, vol. 1, pp. 48–54. ACL, Stroudsburg (2003)
8. Axelrod, A., Mayne, R.B., Callison-burch, C., Osborne, M., Talbot, D.: Edinburgh system description for the 2005 iwslt speech translation evaluation. In: Proc. International Workshop on Spoken Language Translation, IWSLT (2005)
9. Mendling, J., Reijers, H.A., Recker, J.: Activity labeling in process modeling: Empirical insights and recommendations. Inf. Syst. 35(4), 467–482 (2010)
10. Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In: Proceedings of the 5th Annual International Conference on Systems Documentation, SIGDOC 1986, pp. 24–26. ACM, New York (1986)
11. Banerjee, S., Pedersen, T.: An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. In: Gelbukh, A. (ed.) CICLing 2002. LNCS, vol. 2276, pp. 136–145. Springer, Heidelberg (2002)
12. Eid-Sabbagh, R.-H., Kunze, M., Meyer, A., Weske, M.: A Platform for Research on Process Model Collections. In: Mendling, J., Weidlich, M. (eds.) BPMN 2012. LNBIP, vol. 125, pp. 8–22. Springer, Heidelberg (2012)
13. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on ACL, ACL 2002, pp. 311–318. ACL, Stroudsburg (2002)
14. Agarwal, A., Lavie, A.: Meteor, m-bleu and m-ter: evaluation metrics for high-correlation with human rankings of machine translation output. In: Proceedings of the Third Workshop on StatMT, StatMT 2008, pp. 115–118. ACL, Stroudsburg (2008)
15. Becker, J., Delfmann, P., Herwig, S., Lis, L., Stein, A.: Towards Increased Comparability of Conceptual Models - Enforcing Naming Conventions through Domain Thesauri and Linguistic Grammars. In: ECIS 2009 (June 2009)
16. Koschmider, A., Blanchard, E.: User assistance for business process model decomposition. In: First IEEE International Conference on Research Challenges in Information Science, pp. 445–454 (2007)
17. van der Vos, B., Gulla, J.A., van de Riet, R.: Verification of conceptual models based on linguistic knowledge. Data and Knowledge Engineering 21(2), 147–163 (1997)
18. Breuker, D., Pfeiffer, D., Becker, J.: Reducing the variations in intra- and interorganizational business process modeling - an empirical evaluation. In: Proceedings of the Internationale Tagung Wirtschaftsinformatik (2009)
19. Carpuat, M., Wu, D.: Improving statistical machine translation using word sense disambiguation. In: EMNLP-CoNLL 2007, pp. 61–72 (2007)
20. Chan, Y.S., Ng, H.T.: Word sense disambiguation improves statistical machine translation. In: 45th Annual Meeting of the Association for Computational Linguistics, ACL 2007, pp. 33–40 (2007)
21. Nießen, S., Ney, H.: Statistical machine translation with scarce resources using morpho-syntactic information. Comput. Linguist. 30(2), 181–204 (2004)
22. Carbonell, J., Klein, S., Miller, D., Steinbaum, M., Grassiany, T., Frey, J.: Context-based machine translation. In: Proceedings of the 7th Conference of the AMTA, pp. 8–12 (2006)

A Theoretical Basis for Using Virtual Worlds as a Personalised Process Visualisation Approach

Hanwen Guo, Ross Brown, and Rune Rasmussen

Information Systems School, Science and Technology Faculty,
Queensland University of Technology, 2 George St, Brisbane, 4000, Queensland, Australia
{hl.guo,r.brown,r.rasmussen}@qut.edu.au

Abstract. Communication processes are vital in the lifecycle of BPM projects. With this in mind, much research has been performed into facilitating this key component between stakeholders. Amongst the methods used to support this process are personalized process visualisations. In this paper, we review the development of this visualization trend, then, we propose a theoretical analysis framework based upon communication theory. We use this framework to provide theoretical support to the conjecture that 3D virtual worlds are powerful tools for communicating personalised visualisations of processes within a workplace. Meta-requirements are then derived and applied, via 3D virtual world functionalities, to generate example visualisations containing personalized aspects, which we believe enhance the process of communication between analysts and stakeholders in BPM process (re)design activities.

Keywords: Business Process Management, 3D Virtual Worlds, Process Communication.

1 Introduction

Presently, Business Process Management (BPM) is recognized as a set of approaches that can facilitate business activity management [1], and its lifecycle involves four phases, which are design, configuration, enactment and diagnosis [2]. In these four phases, the communication process has been recognized as a vital component, since it is one factor, among others, that is highly related to the success of BPM projects [3].

To enhance the communication process, visual assistance approaches have been employed. Conceptual diagrams, such as BPMN [4] or ER diagrams [5], are shown to other stakeholders by business analysts, when complex business environments need to be investigated. Despite these approaches, it is reported that business analysts and stakeholders often have communication problems [6-8]. This is because we cannot guarantee that every stakeholder has the necessary process diagram knowledge to understand what is being presented [9].

A 3D virtual world is a network-based, computer synthesized dynamic environment, where participants can communicate with each other and interact with rich 3D computer-generated environmental objects [10]. Recently, it has been realized that such 3D virtual worlds can be applied in observational ethnography, case

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 229–240, 2013.

studies, economic markets and social networks [10]. Its richer visualization abilities can expand the memory bandwidth of people, allowing them to effectively process more information [11]. This strongly suggests that 3D virtual worlds could be a superior process visualization platform, enabling people to recall and cognate about conceptual and non-conceptual content. Such an approach may facilitate communication processes in analysing, modelling and validating organizational structures, and related resource behaviours. In addition, these visualization have the capacity for high levels of personalized configuration, providing role-based representations that enable people to recognize their place in a complex process system, via provision of a visual context simulating their own experience [45].

As a new research area, researchers are working on BPM oriented 3D virtual world applications [12-17]. Despite the fact that these works are available and can be recognized as visualization approaches for enhancing communication approaches, theory-based analysis outputs are still lacking. Theory analysis in Information Systems or BPM domains, which is conducted ahead of system developments, can provide guidance that caters to needs in reality [18]. In other words, the lack of theory analysis may impede us in developing sophisticated 3D visualization systems for BPM in the future.

With these points in mind, we propose a theory-based framework and utilize this framework to analyse problematic areas in communication processes, explain why problems occur, predict how to overcome these problem and suggest how to design an effective communication approach within a BPM context.

This paper is organized as follows: Section 2 discusses related work. Section 3 explores the capabilities of using a virtual world as a communication approach. Section 4 provides a mapping of virtual world capabilities to personalized visualizations. And at last, Section 5 concludes with a discussion of achievements, and points towards future work.

2 Related Work

In 2008, we proposed a BPM oriented virtual world application [13], where we claimed that 3D virtual worlds can be utilized as a communication approach in the BPM context. Then, we proposed an extended business animation system [12]. Following these initial efforts, other researchers have published BPM oriented virtual world applications. Perkins [14] advocates that intelligent agent systems should be integrated for demonstrating designed business plans. Chodos et al. [15] have proposed a workflow based simulation system, named MeRiTS, for professional medical training. These works have sought to visualize the operational levels of a business process environment, typically for training applications in the health domain, see Fig. 1. However, along with other commercial systems, such as Flexsim[1], there is an absence of a focus on personalised forms of visualization, in particular, the overlaying of surrounding information aligned with the tasks being represented for a particular business role and its potential personalized variations.

[1] `http://www.flexsim.com/flexsim/`

Fig. 1. Illustrations from previous work by [19] (picture A), [14] (picture B) and [15] (picture C)

One of the major positive findings with virtual worlds has been the quantitative results showing the superior capabilities of such technologies in knowledge transfer in training scenarios [46], or that they are at least equal to present physical methods [47]. This is consistent with theoretical models presenting a proposed hierarchy of simulation above other less interactive approaches, producing better learning outcomes via greater retention of information [48]. These promising initial educational results are highly relevant to process communication issues. Such capabilities in the transfer of knowledge can be seen as overlapping with activities in process redesign requiring clear communication of process knowledge. Evidence has emerged indicating that a more human interaction oriented approach may offer better communication results [41].

Personalised process model visualization and presentation continues to be a very active area of research [42- 45]. Personalized process model visualization modifies a base process model to conform to the specific information requirements of a stakeholder within a process model. Of particular interest is the insight that the use of iconic or pictorial representations of process activities may provide better insight into the process for a stakeholder [45]. Only preliminary evidence is available at the moment to show that this personalized view of a process actually increases understandability [44]. The question of the effectiveness of personalized views is yet to be fully answered in the affirmative.

At this moment in time, there is no research that we are aware of that has explored the possibility of using 3D virtual worlds as an environment for personalized process displays. Therefore, as we believe this is a first paper on this personalized topic, there is a need to provide theoretical analysis motivating such research. In the following, we will propose our analysis framework, with extensions to our previous work [20], in order to support our proposal that a 3D virtual world is a valid and effective approach to personalized process visualization.

3 Framework Development

Our intention in this section is to generate an appropriate analysis framework to support our conjectures regarding the efficacy of 3D virtual worlds for personalized process visualisations. Therefore, using an approach drawing from the popular theory development work of Gregor [18], and used in closely related visualization work [8], we propose an analysis framework with an objective to support, in theory, the argument that a virtual world can be used to improve communication between business analysts and stakeholders in a BPM communication context. The framework involves three major theory development theories from Gregor (Analysis, Explanation and Prediction, and Design and Action) that build on each other to support our conjecture. The analysis theory seeks to characterize the information transmission between business analysts and other stakeholders. The explanation and prediction theory seeks to explain related cognitive processes in the mind of business analysts and other stakeholders, predicting how to enhance them. A final design and action theory is used to support the conjecture that 3D virtual world affordances can support the developed theories on BPM communication.

3.1 Analysis of the BPM Communication Model

Many researchers have created conceptual models and theories for describing such communication. Among these conceptual models, the one proposed by Jakobson [21], Shannon and Weaver [6] has been widely used in different areas, such as information technology [22], software engineering [8] and media studies [23]. The common components in these two models are an information sender and receiver, information coding and transition processes, and factors that can affect the quality of information transition. They [6, 22] both believe good communication requires the sender and receiver to use the same code, while noise, as variations in the coding process, can slow down cognitive processes. The core ideas and components of their conceptual communication models has been adapted to represent communication between business analysts and other stakeholders in a BPM redesign context in Fig. 2. For this context, we concentrate on the business analyst and other stakeholder relationships as a first point of call. The stakeholders in our case are typically domain experts experienced with the processes in question, but not knowledgeable of process model grammars. For example, in a hospital management scenario, the stakeholder may be a nurse, doctor or an administrator.

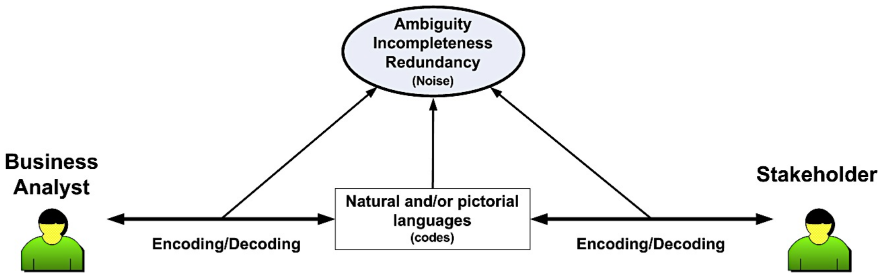


Fig. 2. Illustration of a modified communication model within the BPM context, with the business analyst and other stakeholder taking alternating sender and receiver roles. *Codes* in this case are defined as the pictorial languages used in process modeling, while *noise* has been defined for this context as ambiguity, incompleteness and redundancy in the process representation, leading to reported communication problems between analysts and other stakeholders [9].

In practice, the aim of the diagnosis phase in the BPM life cycle is to capture, analyse, automate and optimize present business processes. Issues such as, measuring the gap between current business processes and desired business processes, should be discussed in this phase [3, 24]. An accurate and mutual understanding between business analysts and stakeholders is therefore critical in decision making regarding process improvements [3].

On the one hand, business analysts need to gain a better comprehension of requirements and feedback from their stakeholders. This can help them design an improvement plan that suits the needs of their clients, and to continually revise the plan according to stakeholder preferences. On the other hand, stakeholders need to articulate their requirements so that business analysts can examine the validity of the newly designed business process model, to confirm that the improved and optimized business processes are appropriate, as proposed by the business analyst.

It is typical, during the design of a visualization approach, to include an analysis of the particular stakeholder tasks needing visual support [49]. We apply this approach to the analyst stakeholder interactions during a process validation activity. During their discussions about a business process, both business analysts and operational managers are interested in a number of topics, with some of them being: the specific sequence of task events, personnel arrangements and human resource behaviours at the operational level of a business environment [25-27]. A diagram showing the communication skill set of a business analyst, and the interests of a manager or client, are represented in Fig. 3, below.

So far, communication theories [6, 22] have been used to analyse the communication roles of business analysts and stakeholders, and the code forms they utilize. It can be said that business analysts and stakeholders alternatively play the role of message sender and receiver in the communication process.

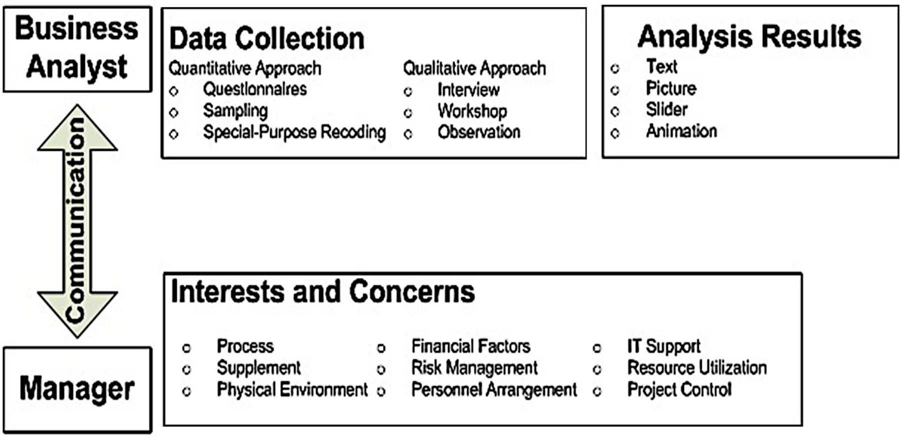


Fig. 3. Illustration of the main information concerns of business analysts and managers

From this previous research, two major conclusions can be reached. One, stakeholders cannot always represent their understanding and thoughts about business activities in a well structured way [7]. Their goal may not be clearly defined and a variety of unexpected factors can influenced their satisfaction with meeting results [28]. Two, the visual codes used by business analysts inevitably interfere with cognitive processes in the readers. To understand the visual codes requires a cognitive process within the stakeholder, to work on the conceptual model representations. The success of the cognitive process relies on long term memory that contains prior knowledge about the conceptual model, such as abstract model grammars and terminologies [8]. In practice, it cannot be expected that visual code readers, in particular domain specialists, have such prior knowledge. Such lack of knowledge leads to results showing that conceptual models puzzle domain specialists [7-9]. In part, this is due to the their inability to align the abstracted representation with their own personal and experiential understanding of the business process being discussed.

3.2 Explanation and Prediction

In the following, conjectures drawn from cognitive theories [11, 29-31] are made to explain why the noise occurs in the communication process and predicts what measures can reduce the noise to promote the efficiency of the communication process.

Explanation. Fig. 4. represents information processes in the human brain according to a popular model known as the Human Process Model (HPM) [32].

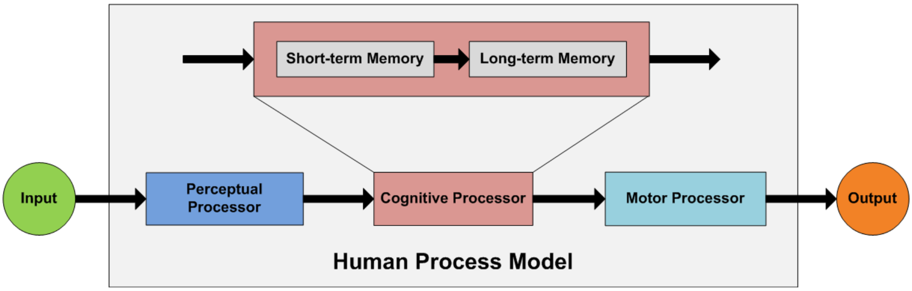


Fig. 4. Illustration of the Human Process Model that is adapted from [32]. The model describes the procedure that humans use to process information.

The Perceptual Processor models how people sense input information, such as a process model in a graphical representation. Sense organs, such as eyes and ears, receive input information, which will be passed to the Cognitive Processor.

The Cognitive Processor in the model, analyses the input information by utilizing short-term memory, which is a temporary storage area loaded with a stream of input signals. It is believed that the loading capability of such working memory is important in both verbal and visual information processing [33, 34].

Long-term memory permanently stores some prior knowledge. The amount of knowledge determines how a human responds to the input information [35]. If prior knowledge is relevant to the input information, the motor processor will receive positive indications, enabling a human to take a correct action. Otherwise, the motor processor will receive negative indications, giving an irrelevant or wrong answer.

With reference to this model of information processing, it can be said that the cognitive processor plays an important role in problem solving [29, 36]. In the context of this research, it can be said that the performance of communication between business analysts and stakeholders is dependent on this same cognitive process.

Therefore, a reverse argument can be made; the increase of loading capability in the short-term memory and presence of prior knowledge in the long-term memory can optimize and improve the cognitive processes involved, potentially improving communication between stakeholders in a meeting.

Prediction. According to cognitive load theory [30], humans have limitations in processing information, that is, humans usually can process seven, plus or minus two items without context [11]. This is a bottleneck with regards to communication improvement and optimization. Miller [11] states that one possible solution for communication improvement is to increase the loading capability of the short-term memory of the receiver, arousing prior knowledge in the longer-term memory to achieve better performance.

In BPM consultation practice, tools and techniques, such as individual or group interviews, questionnaires, conceptual models, workshops and prototypes, aim to achieve better communication [26, 28, 37]. These tools and techniques provide people with a wide range of communication methods.

The main conjecture of this paper is that *personally configurable 3D virtual world tools and techniques may increase the loading capability of the short-term memory and arouse prior knowledge in the longer-term memory*, consistent with the arguments of Miller [11]. For example, an accurate visualization about the working environment may arouse the prior knowledge of a stakeholder. Many stakeholders work in a real environment, not a conceptual space. They are very visually familiar with their working environment. A personalised visualization of their working environment can provide them with a representation in a "hands-on" manner that simulates real artefacts in real spaces. This enables them to make comments according to their work experience.

Moody concludes that this "hands-on" manner is a form of semantic transparency [8]. Such a transparency provides the real meaning and appearance of conceptual models. People can directly infer information in the conceptual model from the operational representation of the conceptual models in a simulated workplace.

Indeed, these visualization capabilities can be combined and employed to inform an individual or group interview, questionnaire, conceptual model, workshop and prototype [26, 28, 37]. Therefore, these visualization needs can be recognized as meta-requirements for communication improvement and optimization, increasing the loading capabilities of the short-term memory to arouse prior knowledge in the longer-term memory.

4 Effective Communication Approach Design

An appropriate Design and Action Theory can guide us to systematically and scientifically design a better communication approach [18]. Two design theories proposed by Markus et al. [38] and Walls et al. [39] state that if system A can meet the meta-requirements for developing another system B, then system A can be adapted to facilitate the development of system B. Guided by such design theories [38, 39], the following sections will investigate the features of a 3D virtual world and then discuss the feasibility of utilizing 3D virtual worlds to satisfy previously mentioned meta-requirements.

Virtual world features can be used to create BPM oriented visualization applications, satisfying communication meta-requirements we have identified, facilitating the BPM Communication Model. The supporting relationship between aforementioned meta-requirements and features of a virtual world application are represented in Table 1.

Table 1. Discussion of the relationship between meta-requirements and virtual world features

Meta-requirements	Virtual World Features
Entity Representation	Geometry Representation
Physical Environment	Geometry Representation
User Participation	Interaction and Collaboration, Avatar
Prototype Demonstration	Geometry Representation, Functional Configuration, Avatars, Behaviour Modelling
Information Display	Functional Configuration, Information Visualization
Human Resource Behaviour	Avatar, Behavioural Modelling

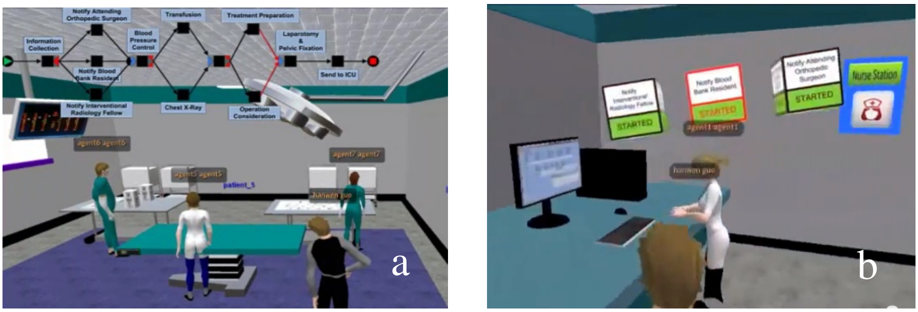


Fig. 5. Example personalized virtual world process visualisations. Firstly, the 2D HUD (a) is a representation of the process model of interest, with its state highlighted using red annotations. The second (b) is a resource-aligned representation providing a role-oriented view of the tasks being specifically performed by the simulation agent within the view.

This process has driven the design of our series of process model simulations and visualisations that can be focused on a particular role within an organization. The visualisations are developed using an agent-based system we have created to simulate control and resource workflow patterns [20]. The agents are provided work using a role-based push approach from a business process system. We now detail the meta-requirements for such communication, and how this has motivated the virtual world visualization design. In particular, we focus on how this can personalize a visualization to assist understanding by a stakeholder. A series of selected features [40] and capabilities are now analysed. In each case, the examples referenced are those shown in Fig. 5, which illustrates a hospital admissions process we have visualized in a virtual world[2].

Entity Representation. Business analysts typically use 2D shapes constrained by visual grammars to abstractly represent real objects, in particular, to describe the relationships between tasks and state transitions between tasks [4]. In the visualisations shown below, the YAWL process modeling grammar is used to annotate the scene to provide process model information. We show two examples of this annotation, via a 2D Heads Up Display (HUD) and a resource centric overhead display. This relates the abstract process activities to an operational environment, with agent representations of specific roles, to increase insight.

Physical Environment. In a 3D virtual world, a complex and realistic environment can be built up from basic geometries. Key physical objects related to processes and roles are depicted in the images as a basis for the visualization. In the case of our example, relevant medical equipment, such as beds, administration desks, amongst other items, are physically modeled in order to provide a representation of a familiar working environment, personalizing the visual to match the workplace of a viewer.

Human Resource Behaviour. Observers can obtain insight into the position level, nature of work, interpersonal relationship, intention and characteristics of an observed person performing work activities via their appearance [14, 25]. In a 3D virtual world, avatars or agents can be created that accurately model the work behaviour of people in reality. In the example shown, tasks have been allocated to agents according to a role model within the process, with their commensurate visual behaviours.

Prototype Demonstration. Business analysts can use some visualization approaches, such as a configurable simulation, to demonstrate a prototype [26, 28, 37]. Such visual assistance is an essential component of requirements elicitation and analysis, enabling business analysts and stakeholders to discuss unforeseen situations that may happen in their work. In a virtual world, a prototype can clearly demonstrate human resource behaviour, enabling stakeholders and business analysts to observe characteristics of the business plan via interactions between agents and objects in the virtual worlds. In a similar vein to configurable process simulation, the examples demonstrate, using multiple agents, each process instance in parallel.

User Participation. Users of virtual worlds can log into a virtual world from a remote location with a software client. The users who log into the virtual world will be represented as avatars, providing people with communication tools for engagement. The darker avatar in each image in the examples, is logged in remotely to the server, and can thus view the simulation as it is performed. Even a side-by-side analyst and stakeholder meeting can benefit from this interaction, as the logged in avatar provides an in-world gesturing device.

[2] Videos of visualizations are available at:
`https://www.youtube.com/watch?v=m6Losma61-U`

Information Display. Information may be loosely classified as qualitative and quantitative. These types of information reflect the temporal state of the business environment. Representations of this information can, amongst other things, provide people with insight into the workload of human resources and utilization rates of non-human resources. Our examples show, via a HUD, the display of the current state of the process model case being visualized, or resource centric information regarding the state of each agent in the simulation.

5 Conclusion

In this paper, a theoretical framework has been proposed for solving problems in communication between analysts and other process stakeholders. Firstly, it has utilized communication theories [6, 21] to identify the main components of communication between business analysts and stakeholders. Then, cognitive theories [11, 29, 30] were used to explain why the performance of the communication process can be negatively affected by more abstract process model grammars. Meta-requirements for communication improvement and optimization were identified. Finally, features of 3D virtual worlds were analysed, forming a mapping of their key features to communication requirements, showing how the features of a virtual world can be combined to assist the validation techniques used by a business analyst. Specific examples from a hospital virtual world we have implemented were used to illustrate the use of virtual worlds for personalized views. In particular, the views can be personalized to the operational view that a stakeholder has of their work.

This work is preliminary in nature, and has only begun to investigate possible approaches to providing 3D personalized visualisations. In future work, we will design an evaluation framework, which is based on the discussion in this paper, to evaluate the presented visualization approaches. In addition, the visualization interfaces will be updated to reflect the latest approaches to 2D personalized visualization [45], providing intuitive filtering and display approaches for a more fine-grained personalized experience of 3D process model information.

Acknowledgments. This research has been supported by Smart Service CRC in Australia (http://www.smartservicescrc.com.au).

References

1. Brocke, J.V., Rosemann, M. (eds.): Handbook on Business Process Management: Strategic Alignment, Governance, People and Culture. Springer, Heidelberg (2010)
2. van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M.: Business Process Management: A Survey. In: van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 1–12. Springer, Heidelberg (2003)
3. Trkman, P.: The critical success factors of business process management. International Journal of Information Management 30(2), 125–134 (2010)
4. Wohed, P., van der Aalst, W.M.P., Dumas, M., ter Hofstede, A.H.M., Russell, N.: On the Suitability of BPMN for Business Process Modelling. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 161–176. Springer, Heidelberg (2006)

5. Davies, I., Green, P., Rosemann, M., Indulska, M., Gallo, S.: How do practitioners use conceptual modeling in practice? Data & Knowledge Engineering 58(3), 358–380 (2006)
6. Shannon, C., Weaver, W.: The Mathematical Theory of Communication. University of Illinois Press, Chicago (1963)
7. Sadiq, S., Indulska, M., Bandara, W., Chong, S.: Major Issues in Business Process Management: A Vendor Perspective. In: 11th Pacific-Asia Conference on Information Systems: Managing Diversity in Digital Enterprises (PACIS 2007), pp. 40–47 (2007)
8. Moody, D.: The "Physics" of Notations: Toward Scientific Basis for Constructing Visual Notations in Software Engineering. IEEE Transactions on Software Engineering 35(6), 756–779 (2009)
9. Khatri, V., Vessey, I., Ramesh, V., Clay, P., Park, S.: Understanding Conceptual Schemas: Exploring the Role of Application and IS Domain Knowledge. Information Systems Research 17(1), 81–99 (2006)
10. Bainbridge, W.: The Scientific Research Potential of Virtual Worlds. Science 317(5837), 472–476 (2007)
11. Miller, M.: The magical number seven, plus or minus two. Psychological Review 63, 81–97 (1956)
12. Brown, R., Rasmussen, R.: Virtual Environment Visualisation of Executable Business Process Models. In: Rao, R. (ed.) Virtual Technologies for Business and Industrial Applications: Innovative and Synergistic Approaches, pp. 68–88. IGI Global Press, Hershey (2010)
13. Brown, R., Cliquet, F.: Communication of Business Process Models via Virtual Environment Simulations. BPTrends 12(9), 1–7 (2008)
14. Perkins, K.: Workflow Simulation in a Virtual World, Master Thesis in Computer Science and Computer Engineering Department, University of Arkansas, Fayetteville (2001)
15. Chodos, D., Stroulia, E., King, S.: MeRiTS: Simulation-Based Training for Healthcare Professionals. Studies in Health Technology and Informatics 163, 125–131 (2011)
16. Betz, S., Eichhorn, D., Hickl, S., Klink, S., Koschmider, A., Li, Y., Oberweis, A., Trunko, R.: 3D representation of business process models. In: MobIS, pp. 73–87 (2008)
17. Eichhorn, D., Koschmider, A., Li, Y., Sturzel, P., Oberweis, A.: Trunko. R.: 3D support for business process simulation. In: 33rd Annual IEEE International Computer Software and Applications Conference, COMPSAC 2009, vol. 1, pp. 73–80 (2009)
18. Gregor, S.: The Nature of Theory in Information Systems. MIS Quarterly 30(3), 611–642 (2006)
19. Brown, R., Rasmussen, R., Baldwin, I., Wyeth, P.: Design and implementation of a virtual world training simulation of ICU first hour handover processes. Australian Critical Care Journal 25(1), 178–187 (2012)
20. Guo, H., Brown, R., Rasmussen, R.: Virtual worlds as a model-view approach to the communication of business processes models. In: CEUR Workshop Proceedings (2012)
21. Jakobson, R.: Closing statement: Linguistics and Poetics, in Style. In: Sebeok, T.A. (ed.) Language, pp. 350–377. MIT Press, Cambridge (1960)
22. Hollnagel, E., Woods, D.: Joint Cognitive Systems: Foundations of Cognitive Systems Engineering. CRC Press, Boca Raton (2005)
23. Manovich, L.: The language of new media. The MIT Press, Cambridge (2001)
24. Nah, F., Lau, J., Kuang, J.: Critical factors for successful implementation of enterprise systems. Business Process Management Journal 7(3), 286–296 (2001)
25. Lewis, M., Slack, N.: Operations Management: Critical Perspectives on Business and Management. Routledge, New York (2003)
26. Reid, R., Sanders, N.: Operations Management. Wiley, New York (2009)

27. Voss, C., Tsikriktsis, N., Frohlich, M.: Case research in operations management. International Journal of Operations & Production Management 22(2), 195–219 (2002)
28. McConnell, S.: Rapid Development: Taming Wild Software Schedules. Microsoft Press, Redmond (1996)
29. Newell, A., Simon, H.: Human Problem Solving. Prentice Hall, Upper Saddle River (1972)
30. Chandler, P., Sweller, J.: Cognitive Load Theory and the Format of Instruction. Cognition and Instruction 8(4), 293–332 (1991)
31. Flugel, J.: The psychology of clothes. AMS Press, New York (1976)
32. Card, S., Moran, T., Newell, A.: The Model Human Processor: An Engineering Model of Human Performance. In: Boff, K.R., Kaufman, L., Thomas, J.P. (eds.) Handbook of Perception and Human Performance, vol. 2, pp. 1–35. Wiley, New York (1986)
33. Caplan, D., Waters, G.: Verbal working memory and sentence comprehension. Behavioural And Brain Sciences 22, 77–126 (1999)
34. Lohse, G.: The Role of Working Memory in Graphical Information Processing. Behaviour and Information Technology 16(6), 297–308 (1997)
35. Kosslyn, S.: Graphics and Human Information Processing. American Statistical Association 80(391), 499–512 (1985)
36. Sweller, J.: Cognitive Load During Problem Solving: Effects on Learning. Cognitive Science 12(2), 257–285 (1988)
37. Cadle, J., Paul, D., Turner, P.: Business Analysis Techniques 72 Essential Tools for Success. British Informatics Society Ltd., Chippenham (2010)
38. Markus, M., Majchrzak, A., Gasser, L.: A Design Theory For Systems That Support Emergent Knowledge Processes. MIS Quarterly 26(3), 179–212 (2002)
39. Walls, J., Widmeyer, G., Sawy, O.: Building an Information System Design Theory for Vigilant EIS. Information Systems Research 3(1), 36–59 (1992)
40. Burdea, G., Coiffet, P.: Virtual Reality. Wiley, New York (2003)
41. Kock, N., Danesh, A., Komiak, P.: A discussion and test of a communication flow optimization approach for business process redesign. Knowledge and Process Management 15(1), 72–85 (2008)
42. de Leoni, M., van der Aalst, W.M.P., ter Hofstede, A.H.M.: Visual support for work assignment in process-aware information systems. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) BPM 2008. LNCS, vol. 5240, pp. 67–83. Springer, Heidelberg (2008)
43. Zhao, X., Liu, C., Sadiq, W., Kowalkiewicz, M.: Process view derivation and composition in a dynamic collaboration environment. In: Meersman, R., Tari, Z. (eds.) OTM 2008, Part I. LNCS, vol. 5331, pp. 82–99. Springer, Heidelberg (2008)
44. Koschmider, A., Reijers, H., Dijkman, R.: Empirical Support for the Usefulness of Personalized Process Model Views. In: Christian, D., Field, M.,S. (eds.) Computer Science Industry Multi Conference, MKWI 2012 (2012)
45. Kolb, J., Kammerer, K., Reichert, M.: Updatable Process Views for User-Centered Adaption of Large Process Models. In: Liu, C., Ludwig, H., Toumani, F., Yu, Q. (eds.) Service Oriented Computing. LNCS, vol. 7636, pp. 484–498. Springer, Heidelberg (2012)
46. Jarvis, S., de Freitas, S.: Evaluation of an immersive learning programme to support triage training. In: Games and Virtual Worlds for Serious Applications, VS-GAMES 2009, pp. 117–122 (2009)
47. Dev, P., Youngblood, P., LeRoy Heinrichs, W., Kusumoto, L.: Virtual worlds and team training. Anesth Clin. 25, 321–336 (2007)
48. Dale, E.: Audiovisual methods in teaching. Holt, Rinehart and Winston, New York (1969)
49. S.M., Meyer, M., Munzner, T.: Design Study Methodology: Reflections from the Trenches and the Stacks. IEEE Trans. Visualization and Computer Graphics 18(12), 2431–2440 (2012)

A Research Program for Studying the Impact of Process Representation on Risk Analysis

Hamzah Ritchi[1] and Jan Mendling[2]

[1] Universitas Padjadjaran, Jl.Dipatiukur 35, Bandung, 40132, Indonesia and Humboldt Universität zu Berlin, Unter den Linden 6, D-10099 Berlin, Germany
hamzahritchi@fe.unpad.ac.id
[2] Wirtschaftsuniversität Wien, Augasse 2-6, A-1090 Vienna, Austria
jan.mendling@wu.ac.at

Abstract. Business processes modeling plays an important role in helping organizations analyze and implement existing business processes. Specifically, business process understanding is an essential aspect for conducting risk assessment and for detecting internal control weaknesses. In current risk assessment practice, a broad spectrum of notations is used to capture processes with relative strengths and weaknesses. These notations range from pure text-based to purely visual diagrammatic formats. This gives rise to the question whether any of these notations should be preferred in the specific audit and accounting information system domain in order to provide better analysis results. Given the mixed results from prior research, this paper aims to establish a theoretical basis for discussing this question. Based on cognitive research, we identify propositions and derive associated hypotheses. Furthermore, we discuss how analysis performance can be measured in an audit context.

Keywords: Business process models, risk and control, audit, BPMN, text.

1 Introduction

Representing business processes in the form of a model has evolved into a primary focus in information system research and practice. Practitioners conduct process modeling for various purposes, among others, to document organizational processes and to specify information system requirement [1], to conduct process improvement, understanding and communication between participants [2], and to provide specification of an executable automation or semi automation workflow [3].

Understanding business processes becomes increasingly important in risk-based audit and other attestation programs. The International Federation of Accountants's International Standard on Auditing (IFAC's ISA) places stronger emphasis on understanding business processes pertinent to financial position of entities [4], which is crucial for subsequent risk identification tasks. Moreover, as organizational processes are increasingly implemented through information systems, the risks embedded in those processes must be considered when performing risk assessment relevant to potential misstatement in the financial accounts. Technically, this requires

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 241–252, 2013.

the identification, analysis, and measurement of risks of material misstatement as well as linking business risks with associated controls. A proper documentation of the company's processes on a detailed level therefore provides a suitable foundation towards the analysis of the company's risk situation [5].

In a more general context, the importance of information presentation in conceptual models has been studied for quite some time. These works involve comparing external representation alternatives in an attempt to identify the most effective format for judgment and decision-making tasks [6]. Despite ample of studies that have been carried out both theoretically and empirically on external presentation [7,8,9,10], there is hardly consensus on the potential superiority of two opposing extremes of representation, namely text and diagrams. Prior studies in accounting information systems (AIS) and auditing showed mixed results when comparing both. These works come with varying degrees of support on whether using diagrammatic representations can actually outperform textual representations for different tasks [11,12,13,14]. This gives rise to the question whether business process models, as opposed to text, should be preferred in risk analysis. Given the importance of business process understanding for the effective risk assessment and the lack of studies comparing process diagrams versus text, the investigation of how best to represent business processes for risk assessment decision-making is imperative.

This research in progress paper provides the conceptual model of a study on whether a model can help users to better understand the underlying processes to support their risk-related decision-making tasks. Specifically, we aim to study search, recognition, and inference processes when using alternative representations. This study is expected to extend prior research into process model understanding towards the AIS audit domain. We build on the theoretical analysis of representational equivalence [7], the Cognitive Fit theory [15], and the Cognitive Load Theory [16].

The remainder of this paper is organized as follows. The next section provides an overview of the background literature with an emphasis on risk assessment of business processes in assurance practice. Then, the next section explains process model use for audit related judgment and decision-making. Afterwards, we define the research model and the proposed propositions, before concluding the paper.

2 Related Works

In this section, we discuss business process risk analysis and specific applications of process modeling for risk analysis.

2.1 Business Process Risk Analysis

The risk-based audit approach places emphasis on the broader business scope rather than directly focusing on financial misstatement. Risk analysis entails identifying relevant business risks of the underlying business processes (risk identification) as well as controls corresponding to those risks, while making decision based on assessing the magnitude of identified risks for any necessary subsequent tests (risk assessment) [17]. Risk assessment conveys the decisions auditors should make as the result of the educing risk information. An accurate representation of business processes therefore helps to understand them better in risk identification tasks.

The risk-based audit approach starts by initially determining business risk exposing the current AIS under investigation. Such risks, when not adequately addressed with appropriate internal controls, may lead to unfavorable material misstatement in the reported financial statement. Hence, the first step is to establish a set of standard controls to reduce the likelihood of the risks. Next, existing controls and the set of standard controls are compared, while any deficiencies and solutions are identified. Finally, auditors and managers test these controls to verify if they are performing as documented [18]. Against this elaboration, risk-driven audit approach thus requires identifying and documenting business processes and controls associated with financial reporting as prerequisites for risk assessment. The increasing usage of information systems automating business processes also adds to the complexity of the business risks applicable to the process of provisioning accounting information [4].

Risks attributed to these processes thus need to be carefully captured and analyzed. One example is placing a purchase order which is the subset of the procure-to-pay business process. A company bears risks of having unauthorized purchase requisition or and paying for a not existing supplier which violates an occurrence or existence assertion. This can be addressed by defining a control objective: that purchase orders are placed only for approved requisitions and existing suppliers. In line with this explanation, the International Standard on Auditing (ISA) 315 § A84 emphasizes the importance to understand information system and business processes [4]:

> "Obtaining an understanding of the entity's business processes, which include how transactions are originated, assists the auditor to obtain an understanding of the entity's information system relevant to financial reporting in a manner that is appropriate to the entity's circumstances."

System applications have built-in controls to ensure information integrity of the related business process. Some of these programmed controls may be critical to the evaluation of internal control over financial reporting. It is therefore essential that auditors acquire sufficient knowledge of all automation technologies of business process execution to evaluate the risk exposures and internal control efficacies.

2.2 Business Process Modeling for Risk Analysis

Previous studies have investigated the importance of representation for analysis tasks. Many of these works compare external representation alternatives in an attempt to identify the most effective format for judgment and decision-making tasks [6]. Performance differences have been studied between different representations such as tables versus graphs [19,20], different text groupings [21,22], and between graphs [10, 23]. Some works postulate a superiority of using diagram for externalizing problem [7,11], while others find no clear superiority [12,24]. Notwithstanding the mixture of these studies, to the best of our knowledge, little attention has been paid on two contrasting alternatives, namely diagrams and text. Such a comparison is highly relevant as risk analysis practice works with the format that is provided by the client.

The theoretical analysis developed by Larkin and Simon [7] emphasizes that diagrams might be superior to text due to their geometrical and topological characteristics. Text representations arguably do not acquire such characteristics. These characteristics are believed to evoke computational efficiency because data structure, programs and attention management system of a production (diagram) enable users to use search, recognition and inference processes to solve problems [7]. From an empirical perspective, an experimental work was performed comparing performance differences in terms of accuracy, time, and user perception between entity relationship (ER) model and Backus-Naur-Form (BNF) language grammar [11]. In that study, ER models were found to be superior in terms of time, but no differences found for accuracy measures. Another research identifies characteristics from literature that are relevant to the situation of when a diagram appears to be superior over text format [24].

The business process modeling literature focuses on the process model as a visual representation of a process [25]. Research in audit risks and controls typically concentrate on procedural aspects such as methodology, documents, or higher-level requirement [see 26,27], while leaving information extraction less explored and open. For instance, how information is represented to facilitate user understanding on business processes receives hardly any attention.

Prior works take two approaches for understanding business processes. The first group can be labeled design-oriented. Design-oriented approaches look at notational aspects of risks and controls that should be introduced in a new, or embedded in an existing model. For instance, Carnaghan [28] compares different process models based on their capacity to represent internal controls in a process level risks assessment. She concludes that current models do not sufficiently address the representational requirements. Furthermore, ontological deficits are discussed for flowcharts and data-flow diagrams in this context [31]; extending ER diagrams with risks and controls [29]; complementing EPCs with risks models [30]; or by mapping out REA and DFD diagrams to identify internal control [18]. None of the aforementioned works draws attention to the merit of process models over text.

The second groups, labeled as behavioral-oriented, typically concentrate on the relationship between graphical notations, user characteristics and decision-making tasks. Research method that is commonly adopted for this stream is empirical work. The works included in this group are [8, 10, 11, 12, 32]. Dunn and Grabski [8] examine the effects of the format (REA vs DCA) for decision-making and found that when there is a high degree of localization, the influence of user experience is decreased. The assessment of performance across four different models in an AIS setting indicates that diagrams are best suited for process oriented tasks [10].

Among the relatively few papers in diagram versus text for risks assessment, it appears that no strong conclusion can be derived whether a diagram surpasses text in decision-making tasks. Boritz et al. [12] report that no difference was found in risk assessment based on either a business process diagrams or a narrative, while an experiment comparing conceptual models with text partially shows the superiority of diagrams over narrative [11]. Interestingly, information arranged around the business processes is proved to be more effective than arranging information cues in other category of groupings such as based on control objectives or transactional codes [21,22]. This may indicate that users might benefit from models since information cues are easily located, thereby enabling proses search, recognize and inference [7]. Again, given this mixed conclusion, the question whether business process model does have the influence on risk assessment remains open.

3 Theoretical Foundations and Hypotheses

This section discusses theoretical foundations and the research model of the work.

3.1 Theoretical Background

In the context of our research problem, at least three theoretical perspectives are relevant. The first relates to the question of comparability, which is discussed by the Theory of Equivalence of Representations [7]. When two representation alternatives are equally inferable, they are informationally equivalent. One representation (diagram) may outperform another one (textual) if it is informationally equivalent, and additionally can facilitate inference in an easier way than another representation, indicating the existence of a computational advantage [7]. A cognitive computational advantage depends upon the orchestration quality between the representation form, the mental programs, and the attention management system, in which the programs the search, recognition and inference are processed. All the three are relevant in a risk analysis setting. Search uses the attention management system to locate relevant elements in a data structure. Recognition involves matching information from the representation with the problem [11]. This is likely to be highly dependent upon representation. Inference is rather a task dependent process, in that it requires a user's domain knowledge to combine information deduced from problem tasks with relevant information stored in the user existing knowledge.

The next relevant theory in this context is Cognitive Fit Theory [15]. Cognitive fit theory states that a correspondence between tasks and information representation leads to superior task performance. Since its introduction, the theories' external validity has been supported through various empirical works involving enhancement in contexts, constructs involved, and modification. General findings consistently state that different kind of representations may or may not fit well with certain tasks [20,34,35,36]. The basic formulation of Cognitive Fit Theory describes processes which are initially operating with respect to the problem representation and to the problem-solving task, to produce the mental representation which then produces a problem solution. Cognitive fit occurs when information type emphasized in both representation and task elements match, resulting to a consistent mental representation for facilitating problem-solving process. When the information type emphasized in both representation and elements do not match, the problem solver must exert additional cognitive effort to either transform the problem representation to better match the task or transform their decision processes to better match better the problem representation, which will hinder performance [6].

The third theory, Cognitive Load Theory (CLT) defines the cognitive constraints associated with humans [9]. Humans as problem solver have a limited number of information chunks to process in their short-term memory, and "…any problem that requires a large number of items to be stored in short-term memory may contribute to an excessive cognitive load" [16]. Following the theory, there are three sources of cognitive load: intrinsic load, extraneous load and germane load [16]. While intrinsic load is inherently attached to the task complexity (determined by the elements interactivity and the processing procedure), an unfavorable effect on learning is more attributed to extraneous load when the task and problem information presented force a learner to spend more mental efforts than what the working memory can handle. Appropriate manipulation of information representation should reduce extraneous load and instead promote the germane load, which contributes to schema construction, therefore facilitating learners learning.

3.2 Research Model

Against the theoretical background elaborated above, we develop a research model as the framework for structuring our research, and accordingly set out propositions pertinent to which representation is best to facilitate business process understanding for risk assessment. Research models operationalization set up in [9,14] is adopted and synchronized with the current study. Figure 1 shows our research model.

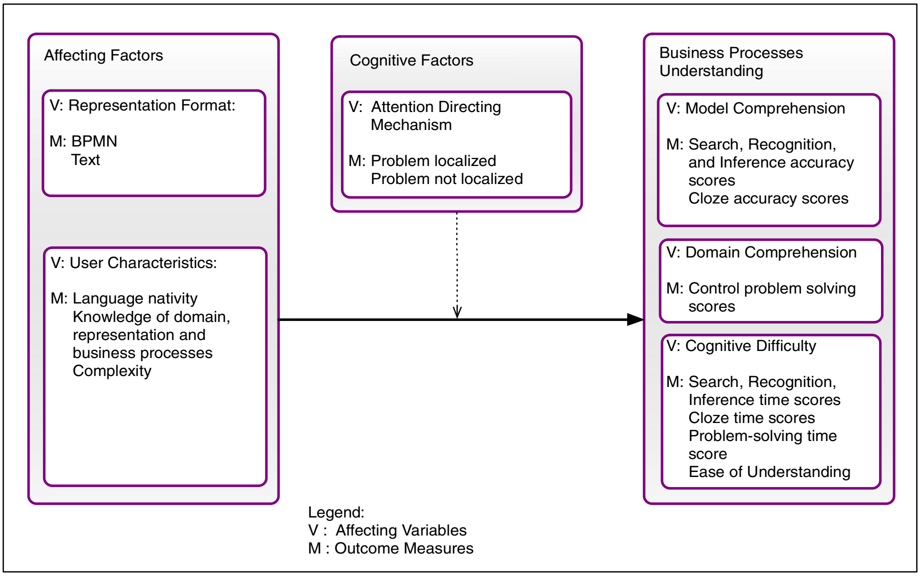


Fig. 1. Research Model

As theorized in [7], we expect that adopting diagrams would better facilitate business process understanding compared to using textual representation. The advantage appears to be observable through an examination on search, recognition and to some degrees, inference processes run by program on the available data structure supported by attention directing mechanism. Given the varied findings on BPM-narrative comparison, careful investigation of how BPM would lever risk analysis is warranted. We aim to use the Business Process Model and Notation (BPMN) as a typical process notation [37] and text narratives as the other end. We choose BPMN as the alternative model because it is the widely accepted process modeling standard. BPMN is intended be readily understandable by business analysts, system engineers and general business people [39].

The theory of cognitive fit is then revisited to provide the rationale for setting up the necessary propositions. As suggested by the theory, when the form of problem representation matches with the problem tasks, users are expected to have a consistent mental model which further enables them to enhance performance quality. One factor that is an important element of the matching process is the capability to localize a problem in specific groupings on an internal level (cognitive). This localization is an important element of the attention directing mechanism as suggested by [7] and has been elaborated by [8] for REA-DCA tasks comparison. According to [7], diagrams may be advantageous in attention management as opposed to narratives. When one can localize a specific area in a problem representation, the process of matching with problem tasks is enhanced without cognitive load of mental transformation processes.

With this reasoning, we argue that using BPMN would enable users to direct attention better on specific areas of a problem when searching pieces of information, recognizing the relation upon the task, and ultimately inferring better conclusions for particular tasks. In accordance with our research model above, the following propositions are suggested, followed by our proposed hypotheses operationalization:

- P1: A representation that better supports the localization of risk and control problems leads to better performance.

Table 1. Effects of Process Representation Alternatives and Cognitive Fit Situation

<table>
<tr><th colspan="2" rowspan="2">Analysis of Effects
Type of Representation</th><th>Model Understanding</th><th>Problem-Solving Understanding</th><th colspan="2">Cognitive Difficulty</th></tr>
<tr><th>Accuracy</th><th>Accuracy</th><th>Time</th><th>Ease of Understanding</th></tr>
<tr><td>Narrative</td><td>Attention directing mechanism is weak since control issues are hard to localize:
• Cognitive Fit does not exist.
• Requires transformation.</td><td>Lower</td><td>Lower</td><td>More</td><td>Lower</td></tr>
<tr><td>BPMN</td><td>Attention directing mechanism is strong since control issues are easy to localize:
• Cognitive Fit exists.
• Requires less transformation, if any.</td><td>Higher</td><td>Higher</td><td>Less</td><td>Higher</td></tr>
</table>

Table 1 summarizes the development of the effects of alternative representations to the dimensions of business process understanding. For giving better clarity, specific naming for tasks – search, recognition, inference, cloze – is removed. Statement in proposition 1 is then translated further into the following hypotheses:

- H1a: More control issues are identified using BPMN than narrative for search tasks.
- H1b: More control issues are identified using BPMN than narrative for recognition tasks.
- H1c: More control issues are identified using BPMN than narrative for inference tasks.
- H1d: Higher scores are obtained using BPMN than narrative for cloze tests.
- H1e: Higher scores are obtained using BPMN than narrative for problem-solving tasks.

Accordingly, diagrammatic business process representation will provide users with greater computational efficiency than the equivalent text-based counterpart.

The existence of computational efficiency may also lead to a favorable decrease in efforts spent for understanding a model. This means that users might find equivalent number of risk and control cues, but one user is able to do a task relatively faster than others. According to cognitive load theory, this occurrence denotes a reduction in cognitive load, preferably the extraneous load element. The load reduction is reasonably associated with attention directing mechanism that works better to localize information stored in diagrammatic representations than narrative representations. Therefore we have:

- H1f: Users using BPMN will identify control issues when performing search tasks faster than narrative.
- H1g: Users using BPMN will identify control issues when performing recognition task faster than narrative.
- H1h: Users using BPMN will identify control issues when performing inference tasks faster than narrative.
- H1i: Users using BPMN will perform faster when performing cloze tests than narrative.
- H1j: Users using BPMN will perform faster when performing problem-solving tests than narrative.

Task performance is contingent to whether a fit is present between the knowledge emphasized in presentation format and the type of knowledge required to solve the task, hence the cognitive fit. Prior research has shown that organizing information around business process enables users to find more risk and control factors than users who used materials organized around financial statement categories [21,22]. We see that diagrammatic representations arguably have a closer fit with the risk assessment tasks that provide better attention directing mechanism than narratives, although both formats emphasize process flow. We then propose the following hypothesis:

- H1k: The presence of attention directing mechanism is positively associated with the higher control scores, higher efficacy and faster time for all tasks.

Prior accounting and information system research has demonstrated that user characteristics, and or environmental characteristics all contribute to performance [38]. Therefore, we need to take user characteristics into account as influencing factors for business process understanding. Involvement of user characteristics in the research model contributes to answer the pragmatic dimensions of the study such as user knowledge and exposure to both business process modeling and auditing. Putting user characteristics into the analysis, the following proposition is stated:

- P2: User characteristics to process representation will affect users' tasks performances.

The proposition is elaborated more into several hypotheses. First hypotheses predict the effect of the use of English as a second language. This is reasonable since the study incorporates a broad set of diverse participant nationality. [9] advises that English as a second language is one potential impediment in process understanding. By the same token, user perception might be considered as factor that slows down the understanding. The more complex a problem case, in absence of experience, would correspond to an inverse of accuracy and time:

- H2a: The use of English as a second language and perceived complexity will have a negative effect on model understanding accuracy scores.
- H2b: The use of English as a second language and perceived complexity will have a negative effect on domain understanding accuracy scores.
- H2c: The use of English as a second language and perceived complexity will have a negative effect on model understanding completion time.
- H2d: The use of English as a second language and perceived complexity will have a negative effect on domain understanding completion time.

Other potential user characteristics worth considering are knowledge of representation, knowledge of business the case presented in the experiment, and knowledge of the business processes and auditing domain. For simplification purpose, we combine these aspects as the following hypotheses:

- H2e: Higher knowledge in representation, case, and domain will be positively associated with model understanding accuracy scores.
- H2f: Higher knowledge in representation, case, and domain will be positively associated with domain understanding accuracy scores.
- H2g: Higher knowledge in representation, case, and domain will be positively associated with model understanding completion time.
- H2h: Higher knowledge in representation, case, and domain will be positively associated with domain understanding completion time.

Furthermore, different information presentation formats would also affect the perception of ease of understanding. This perception (perceived ease of understanding) indicates the cognitive load users have to bear to complete the task given to them within the specified time frame. [40] suggests using this measure to reflect perception of ease of understanding with different term, perceived difficulty. Therefore, the following proposition is stated:

- P3: Business processes representation that can localize risks and controls problems or provide attention-directing mechanism in particular task representation is perceived to be easier than one that does not.

The corresponding hypothesis is:

- H3: Users perceive BPMN to be easier to understand than text narratives.

The derived hypotheses provide the basis for an empirical research program on studying the influence of process representation on risk analysis performance. Such a program is highly timely given the rather ad-hoc utilization of process-related documentation when analyzing risks in practice. The insights brought forward by this research program are deemed to have strong implications for risk analysis research and practice. Most notably, they might spark efforts to standardize visual representations in the accounting information systems domain.

4 Conclusion

In this paper, we have considered different representational options to support risk analysis in the context of business processes. This research is motivated by increasing risk analysis requirements and regulations, an ad-hoc usage of representations in practice, and contradicting findings on preference for a particular representation format in general research on conceptual models. We have discussed relevant theoretical foundations including information equivalence, cognitive load and cognitive fit in order to establish a research model. This research model and its theoretical foundation provide us with the basis to formulate an extensive set of hypotheses, which together form a research program to study the phenomenon.

The next stage of this work is to conduct various experiments. Careful consideration must be taken in terms of execution control and measurement control. Beside that, its cost with regards to investigation and its ease of replication are high [41]. A challenge in this context will be a balanced involvement of populations that are easy to reach like graduate business students and the group of practitioners that are difficult to reach. We aim for a web-based experiment in order to achieve a better access for the latter group.

References

1. Figl, K., Mendling, J., Strembeck, M.: Towards a Usability Assessment of Process Modeling Languages. In: Nüttgens, M., Rump, F., Mendling, J., Gehrke, N. (eds.) EPK 2009, CEUR Workshop Proceedings, Berlin, Germany, vol. 554, pp. 118–136 (2009)
2. Indulska, M., Green, P., Recker, J., Rosemann, M.: Business Process Modeling: Perceived Benefits. In: Laender, A.H.F., Castano, S., Dayal, U., Casati, F., de Oliveira, J.P.M. (eds.) ER 2009. LNCS, vol. 5829, pp. 458–471. Springer, Heidelberg (2009)
3. Mendling, J.: Metrics for Process Models. Springer, Heidelberg (2008)
4. International Federation of Accountants. International Standard on Auditing 315, `http://www.ifac.org`

5. Karagiannis, D.: A Business Process-Based Modelling Extension for Regulatory MKWI 2008. In: Joint Conference Proceedings Multikonferenz Wirtschaftsinformatik 2008, Munich, February 26-28 (2008)
6. Kelton, A.S., Pennington, R.R., Tuttle, B.M.: The Effects of Information Presentation Format on Judgment and Decision Making: A Review of the Information Systems Research. Journal of Information System 2(24), 79–105 (2010), doi:10.2308/jis.2010.24.2.79
7. Larkin, J.J., Simon, H.A.: Why a Diagram is (Sometimes) Worth Ten Thousand Words. Cognitive Science 11, 65–100 (1987)
8. Dunn, C., Grabski, S.: An Investigation of Localization as an Element of Cognitive Fit in Accounting Model Representation. Decision Sciences 32(1), 55–94 (2001)
9. Recker, J.C., Dreiling, A.: The effects of content presentation format and user characteristics on novice developers'understanding of process models. Communications of the Association for Information Systems 6(28), 65–84 (2011)
10. Jones, R.A., Tsay, J.J., Griggs, K.: An Empirical Investigation of the Cognitive Fit of Selected Process Model Diagramming Techniques. The Review of Business Information System 4(6), 1–10 (2002)
11. Dunn, C., Gerard, G.J.: Auditor Efficiency and Effectiveness with Diagrammatic and Linguistic Conceptual Model Representations. International Journal of Accounting Information Systems 2, 223–248 (2001)
12. Boritz, E., Borthick, A.F., Presslee, A.: Narratives versus diagrams: The impact of alternative business process representations on auditor risk and control assessments. In: American Accounting Association IS Section, Mid-Year Meeting, Atlanta, GA, January 4-6 (2010)
13. Alencar, P.S., Efrim Boritz, J., Carnaghan, C.: Business Modelling to Improve Auditor Risk Assessment: An Investigation of Alternative Representations. In: Annual Conference of the Administrative Sciences Association of Canada 2008, University of Waterloo (2008)
14. Gemino, A.C.: Empirical Comparison of System Analysis Modeling Techniques. PhD Thesis. The University of British Columbia (1999)
15. Vessey, I.: Cognitive Fit: A Theory-Based Analysis of the Graphs Versus Tables Literature. Decision Sciences 2(22), 219–240 (1991)
16. Sweller, J., Merrienboer, J.J.G., Paas, F.G.W.C.: Cognitive Architecture and Instructional Design. Educational Psychology Review 10, 252–296 (1998)
17. Ritchi, H., Mendling, J.: Business Process Models for Risk Analysis: Expert View. In: 2012 International Conference on Information Resources Management, Vienna (2012)
18. Xiong, Y., Martin, M.: Mapping Internal Controls Using System Documentation Tools. The Review of Business Information Systems 1(10), 83–92 (2006)
19. DeSanctis, G., Jarvenpaa, S.L.: Graphical Presentation of Accounting Data for Financial Forecasting: An Experimental Investigation. Accounting, Organization and Society 14, 509–525 (1989)
20. Vessey, I., Galletta, D.: Cognitive fit: An Empirical Study of Information Acquisition. Information Systems Research 1(2), 63–84 (1991)
21. Kopp, L.S., O'Donnell, E.: The influence of a business-process focus on category knowledge and internal control evaluation. Accounting, Organizations and Society (30), 423–434 (2005)
22. O'Donnell, E., Schultz Jr., J.J.: The Influence of Business-Process-Focused Audit Support Software on Analytical Procedures Judgments. Auditing: A Journal of Practice and Theory 2(22), 265–279 (2003)

23. Khatri, V., Vessey, I., Ramesh, V., Clay, P., Park, S.J.: Understanding Conceptual Schemas: Exploring the Role of Application and IS Domain Knowledge. Information Systems Research 17, 81–99 (2006)
24. Alencar, P.S., Boritz, J.E., Carnaghan, C.: The Relative Merits of Diagrammatic versus Textual Representations: A Literature Revew of Theoretical and Empirical Perspectives. Working Paper. University of Waterloo, Waterloo, Canada (2004)
25. Moody, D.L.: The “Physics” of Notations:Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. IEEE Trans. on Soft. Eng. 35, 756–779 (2009)
26. Curtis, M.B., Jenkins, G., Bedard, J.C., Deis, D.R.: Auditors’ Training and Proficiency in Information Systems: A Research Synthesis. Journal of Information Systems 23, 79–96 (2009)
27. Eilifsen, A., Knechel, W.R., Wallage, P.: Application of the Business Risk Audit Model: A Field Study. Accounting Horizon 15, 193–207 (2001)
28. Carnaghan, C.: Business Process Modeling Approaches in the Context of Process Level Audit Risk Assessment: An Analysis and Comparison. Int. J. of Accounting Information System 7, 170–204 (2006)
29. Mansour, R., Murthy, U.S.: Consideration of Risks and Internal Controls in Business Process Modeling. In: Österle, H., Schelp, J., Winter, R. (eds.) Proceedings of the 15th European Conference on Information Systems, University of St. Gallen, St. Gallen, pp. 588–599 (2007)
30. Muehlen, M.Z., Rosemann, M.: Integrating Risks in Business Process Models. In: 16th Australasian Conference on Information Systems, Sydney, November 29-December 2 (2005)
31. Recker, J.C., Rosemann, M., Indulska, M., Green, P.: Business process modeling: A Comparative Analysis. J. of the Association for Information Systems 10, 333–363 (2009)
32. Bradford, M., Richtermeyer, S.B., Roberts, D.F.: System Diagramming Techniques: An Analysis of Methods Used in Accounting Education and Practice. J. of Information Systems 21, 173–212 (2007)
33. Siau, K.: Informational and Computational Equivalence in Comparing Information Modeling Methods. J. of Database Management 15, 73–86 (2004)
34. Shaft, T.M., Vessey, I.: The Role of Cognitive Fit in The Relationship between Software Comprehension and Modification. Management Information Systems Quarterly 1(30), 29–55 (2006)
35. Tuttle, B.M., Kershaw, R.: Information Presentation and Judgment Strategy from a Cognitive Fit Perspective. J. of Information Systems 12, 1–17 (1998)
36. Agarwal, R., Sinha, A.T., Tanniru, M.: Cognitive Fit in Requirements Modeling: A Study of Object and Process Methodologies. J. of Mgmt. Information System 13, 137–162 (1996)
37. Object Management Group. Business Process Model and Notation (BPMN) version 1.2., `http://www.omg.org/spec/BPMN/1.2`
38. Dunn, C.L., Grabski, S.V.: Perceived Semantic Expressiveness of Accounting System and Task Accuracy Effects. Int. J. of Accounting Information System 1, 79–87 (2000)
39. White, S.A.: Introduction to BPMN (2004), `http://www.bpmn.org/Documents/Introduction%20to%` (June 20, 2008)
40. Figl, K., Laue, R.: Cognitive Complexity in Business Process Modeling. In: Mouratidis, H., Rolland, C. (eds.) CAiSE 2011. LNCS, vol. 6741, pp. 452–466. Springer, Heidelberg (2011)
41. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering – An Introduction. Kluwer Academic Publishers (1999)

Toward Innovative Model Based Enterprise IT Outsourcing

Vinay Kulkarni and Sagar Sunkle

Tata Research Development and Design Center, Tata Consultancy Services
54B, Industrial Estate, Hadapsar Pune, 411013 India
{vinay.vkulkarni,sagar.sunkle}@tcs.com

Abstract. There are signs that cost arbitrage model of outsourcing engagement between enterprises and service providers will have to change to one based on value generation. Automation decisions taken by siloized businesses under cost arbitrage models have led to complex cost-ineffective situations. Both these situations may be addressed by an innovative business model based on product family concepts that systematically targets transactional and transformational needs of enterprises with focus on value generation using analysis and operational world views of enterprise IT systems. We motivate and elaborate such an approach. Our contributions are innovation that leads to mutual win-win situation by enabling service providers to service IT needs of multiple enterprises of same vertical and by enabling enterprises to reap value-oriented benefits with analysis and operational world views of IT systems thus serviced.

Keywords: Outsourcing, Enterprise Modeling, Enterprise Architecture, Analysis, Operationalization.

1 Introduction

A large portion of enterprise's operational processes are today automated through the use of IT systems. The as-is state of the enterprise is largely the result of a sequence of automation decisions each taken in specific contexts. These specific contexts come into existence due to siloized nature of businesses [1], which leads to several problems with the enterprise's IT systems such as- a) either partial or complete overlap of business functionality, b) sub-optimal design of operational processes, c) sub-optimal implementation of operational processes, d) plethora of non-interoperable technology platforms, and so on [2]. Given the tight economic environment, enterprises can stay viable only if the IT costs are reduced by a factor. This calls for a whole new approach to service the IT needs of enterprises.

The present headcount based labor and cost arbitrage model through which enterprises engage with IT service providers (SPs) either for managing bottom-line or adding to the top-line seems to have ceased to provide the desired value. We believe that enterprises would want to increasingly focus on their core competencies and look for an IT SP who can be the single source catering to all transactional and transformational IT needs. Enterprises will soon start demanding outcome-based

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 253–263, 2013.

pricing thus making IT SPs share some of the operational risks. Minimally, this necessitates that IT SPs shift from labor-based to asset-based service delivery. Thus, it is becoming imperative that IT SPs invest, safeguard and nurture business assets in a manner that IT needs of a set of enterprises be met with little modification if at all.

Owing to our expertise in model-driven engineering of large enterprise applications [3-6] in cost effective manner [7, 8], we propose a model-driven take on conceptualizing and realizing a new outsourcing business model that enables servicing of transactional and transformational needs of enterprise IT systems with outcome-based pricing and on operational risk sharing basis. The set of interacting IT systems, the technology infrastructure they use, and the hardware infrastructure used for execution together can be viewed as an *IT Plant* for the enterprise. Our key contributions with regards to servicing IT plants of enterprises are twofold- one, we propose a new IT plant family (product-line) driven business model and show that this innovative model can prove beneficial in catering to needs of enterprises thus creating a win-win situation for both SPs and enterprises; and second, we provide two kinds of support to realize the proposed business model namely, analysis and operationalization. While analysis level support provides demonstrable evidence of functional and non-functional characteristics required of the IT plant, operationalization level support plans to use the insights obtained from analysis in the actual implementation of the IT plant.

The rest of the paper is organized as follows. Section 2 presents motivation and outline of the proposed solution. Section 3 elaborates why innovation may be the only way outsourcing of future will take place. In Section 4, we describe how value-oriented benefits apart from cost benefits may be obtained using analysis and operational world views of IT plants. In Section 5, we put forth several work-packages needed in order to realize IT plant family for enterprises in the same vertical and prove beneficial to both enterprises and SPs. Section 6 reviews the key related work. Section 7 concludes the paper.

2 Motivation and Outline

Enterprises use IT systems to derive mechanical advantage through automation of operational processes catering to their strategic, tactical and operational needs [14]. The needs could either be transactional or transformational thus leading to run-the-business (RTB) and change-the-business (CTB) costs respectively. The set of interacting IT systems, the technology infrastructure they use, and the hardware infrastructure used for execution of an IT plant, in addition to people constitute principal levers for managing IT cost to business. For example, cost-arbitrage outsourcing model has brought down costs significantly by transferring development and maintenance of IT systems to low cost geographies i.e. replace enterprise's people with SP's; consolidation and rationalization of hardware infrastructure led to significant cost savings; harmonization of technology infrastructure further accentuated the benefits derivable from hardware consolidation; and transformation of IT system implementation in response to these and other changes has brought down the costs even further.

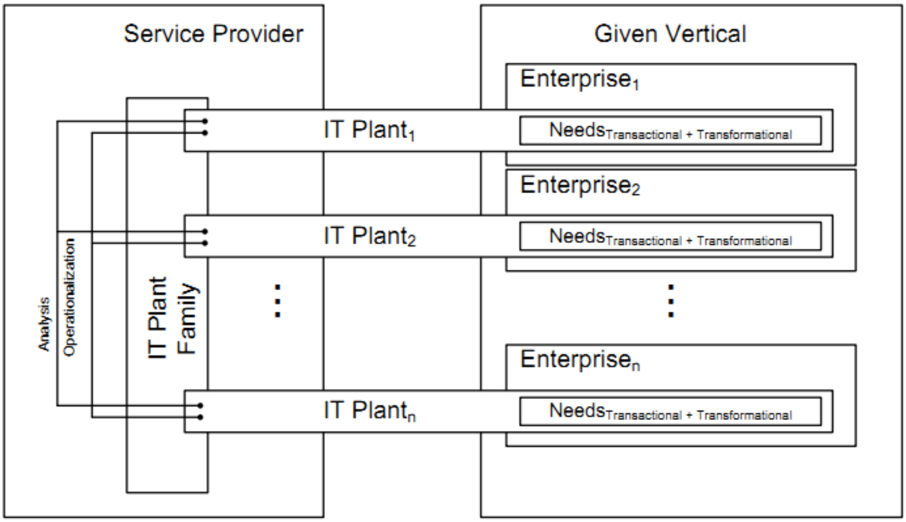


Fig. 1. IT Plant Family Servicing Needs of Multiple Enterprises in Given Vertical via Analysis and Operationalization

Outsourcing and hardware consolidation are fast approaching the point of diminishing returns and harmonization of software infrastructure can bring in only so much benefit [15]. Since an IT system typically comes into existence to service enterprise functional need in local and specific context, any amount to improvement of the IT system is unlikely to guarantee improvement in the IT plant as a whole. Thus, the current practice seems to be approaching its limits in terms of cost effectiveness. It is being realized that beyond the old world labor and cost arbitrage model, an outsourcing model is required that can deliver value based on current business objectives and outcomes. Our proposal in this regard is illustrated in Fig 1.

Achieving scale through common software infrastructure is an important objective for us. Since needs of no two enterprises are likely to be exactly the same, ability to derive implementation of the desired IT plant from its high level specification is necessary. Simplistic strategy of clone-n-own will lead to a plethora of IT plant specifications exhibiting high degree of redundancy (as a result of the commonality) and hence maintenance and evolution problems. Thus, there is a need to manage the related set of IT specifications as a single entity from which the desired specification can be automatically derived under human guidance. This is indicated in Fig. 1 as product line/family concept [8] applied to IT plants for enterprises in same vertical. The key elements of the enterprise IT outsourcing business model and its realization as illustrated in Fig 1 are as follows:

1. It should be possible for an SP to cater to IT needs of multiple enterprises through a single multi-tenant IT plant. It essentially means that an SP can use IT plant specification and implementation in product-line like manner for catering to needs of enterprise in the same vertical. IT plant of an individual enterprise should be easily configurable to meet a priori known specificities of an enterprise.

2. It should be possible for an enterprise to get a feel for the IT plant being offered by an SP both in terms of functional and non-functional characteristics. It should be possible for SP to know the IT needs of enterprise so that they can be demonstrably met by the IT plant being offered.

In the following section, first we elaborate the reasons that necessitate innovation in enterprise IT outsourcing and then describe our proposed solution in detail.

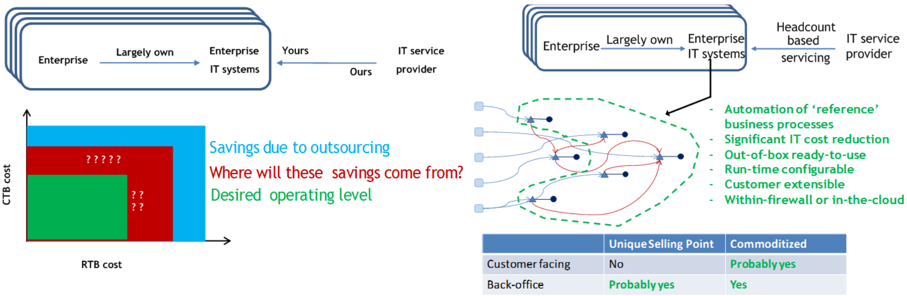


Fig. 2. Costs of Enterprise IT Systems & What is usually automated in an Enterprise?

3 Reasons for Innovation in Enterprise IT Outsourcing

Enterprises have been using IT systems to obtain mechanical advantage through automation of their operational processes. Over the years the dependence of enterprises on their IT systems has been increasing rapidly and there are no signs evident of this changing. Enterprises have traditionally maintained their own IT departments to service IT needs of the business. IT departments have traditionally been viewed as cost centers and evaluated mainly based on RTB and CTB costs to the enterprise.

Outsourcing provided a major lever for reducing RTB costs through replacement of 'your' people (enterprise's) with 'ours' (IT SP's) as shown in Fig 2 (Left). Large enterprises have traditionally been operating in a siloized manner for ease of management and control. As a result, IT needs of an enterprise become apparent *in parts* with a dedicated IT application getting implemented to cater to each *in part* need. To a large extent these needs are of back office with commodity nature as shown in Fig 2 (Right). This leads to plethora of IT applications servicing almost the same IT needs within an enterprise resulting in following problems-

1. Widespread redundancy of IT systems is arguably the single largest cause for highly escalated cost of IT to business.
2. Another side-effect of the same is sub-optimal implementation of business processes that typically spread across many departments. Moreover, current practice of siloized operation makes it very hard for the complete enterprise wide picture depicting its IT systems and how they interact with each other to emerge.

3. In the absence of complete information regarding the as-is state of enterprise's IT systems, the transformational decisions are typically taken putting faith in the ability of *gurus* to predict which of the many possible states will be the most beneficial. Latency of validating such predictions is typically in terms of months if not years and comes with high system development costs that may have to be completely written off.

Faced with these consequences, an enterprise, in absence of the complete picture, has no option but to live with the as-is state that is destined to get increasingly sub-optimal over time as local fixes keep on getting introduced. Increased business dynamics, highly connected nature of IT systems, rapid rate of technology advance / obsolescence, heterogeneity and wide variety of technology platforms used by an enterprise, and sheer number of business applications all further contribute to making the management of CTB costs an involved and almost an intractable problem. Clearly a new approach for managing CTB costs with certainty is needed. We describe our proposal for such approach in the next section.

4 Analysis and Operationalization of IT Plants

We describe our proposal by looking at enterprise IT outsourcing in terms of two related worlds namely, analysis and operational. We propose further that both these worlds can be realized as being model-centric as shown in Fig 3.

Analysis world would represent an enterprise in terms of its goals, operational processes, organizational structure etc. These models are closer to the business domain and yet machine-manipulable so as to be able to establish a specific property and/or explore answers pertaining to questions regarding efficacy of the as-is state, a set of possible states, most desirable among the possible states with regards a given criterion etc.

Though models pertaining to this world might be created from a restricted perspective of IT systems automating the enterprise, they do not model the IT systems per se. Analysis world enables (data-driven) decision making thus reducing dependence on experts' intuition and/or expertise. Analysis world is essentially supposed to come up with interesting possibilities for improving the current state of enterprise and also outline a path from as-is state to the desired to-be state. Enterprises operate in a dynamic environment and hence needs to change continuously along with its IT plant. Response to a change needs to be fast and accurate. To check what happens when a specific response is chosen, models for supporting what-if and if-what scenario playing would be required. These models need to be domain-specific; rather, there is a need to model the domain itself to enable its automated analysis. Analysis world would represent an enterprise in terms of its goals, operational processes, organizational structure etc. These models are closer to the business domain and yet machine-manipulable so as to be able to establish a specific property and/or explore answers pertaining to questions regarding efficacy of the as-is state, a set of possible states, most desirable among the possible states vis-\`a-vis a given criterion etc.

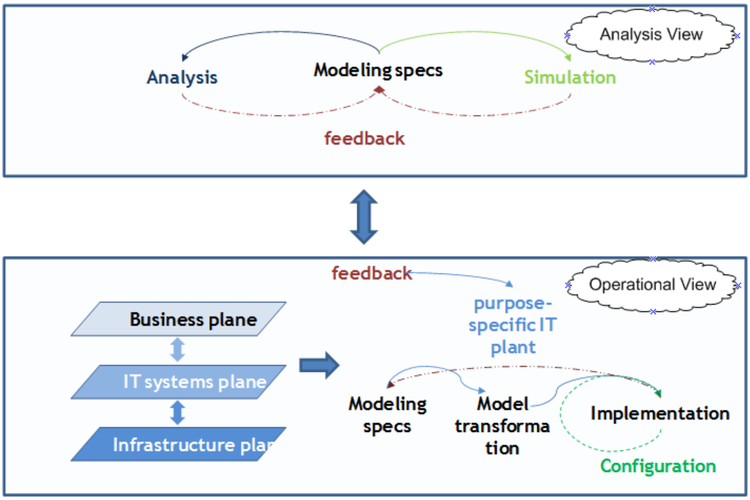


Fig. 3. Analysis and Operational Worlds of Enterprise IT Plants

Though models pertaining to this world might be created from a restricted perspective of IT systems automating the enterprise, they do not model the IT systems per se. Analysis world enables (data-driven) decision making thus reducing dependence on experts' intuition and/or expertise. Analysis world is essentially supposed to come up with interesting possibilities for improving the current state of enterprise and also outline a path from as-is state to the desired to-be state. Enterprises operate in a dynamic environment and hence needs to change continuously along with its IT plant. Response to a change needs to be fast and accurate. To check what happens when a specific response is chosen, models for supporting what-if and if-what scenario playing would be required. These models need to be domain-specific; rather, there is a need to model the domain itself to enable its automated analysis.

Operational world of enterprise IT systems would represent an enterprise in terms of models its business processes, applications used for automating the business process tasks, agents performing the assigned tasks, Technology and IT infrastructure needed for execution etc. In short, a model of the complete IT plant that automates a set of operational processes through a set of software systems ensuring correct operation of the enterprise both in functional and non-functional sense. Since the aim is to cater to the IT needs of multiple enter prises in a given vertical and IT needs of no two enterprises are likely to be exactly the same, purpose-specific IT plant seems required. However, this doesn't lead to a viable business model. As a result, it is imperative that the IT plant actually is a \emph{family} of a set of related IT plants such that a purpose-specific IT plant can be easily derived. Ideally, the derivation process should be as simple as selecting one from the set of many a priori known well-formed and internally consistent configurations. Also, the derivation process should be user-controlled and be effectible at run-time for greater agility. In fact, every element that IT plant comprises of must also be configurable.

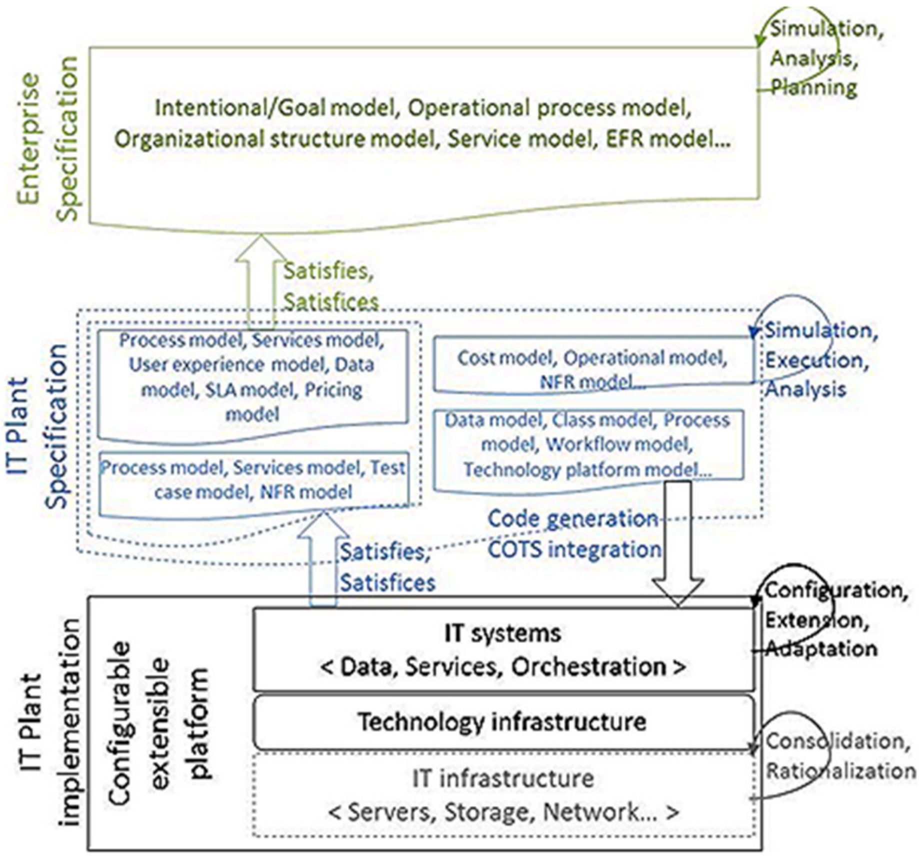


Fig. 4. Schematic Overview

The key challenge pertaining to operational world is that the IT plant should deliver the desired functional and extra-functional requirements and be realizable in terms of multiple technology platforms. The specification of the IT plant should be closer-to-problem-domain, intuitive and technology agnostic so that domain experts will find the notation easy to use. The specification should be complete in terms of its ability to address stakeholder needs on the user side of IT plant. Minimally, it is felt, the specification should cater to functionality, business processes, data and user experience aspects of IT plant and their inter-relationships.

Without a traceable link, possibly bi-directional, between analysis and operational worlds it would be impossible to utilize insights obtained in the analysis world in the operational world. Given the widely different nature of models (and meta models) belonging to analysis and operational worlds, this could be a hard problem. The next section elaborates the research challenges involved in realizing in concert the analysis and operational worlds of IT plant family for enterprise IT systems. Also, solutions to these research challenges can come together in multiple ways each possibly enabling a different opportunity for creating significant business impact.

5 Work-Packages for Realizing Enterprise IT Plants

This entire line of thought is represented in Fig 4 where we present what needs to be done in order to achieve analysis and operationalization of IT plants that address enterprise IT needs in terms of several work-packages:

Enterprise Specification. This work-package constitutes coming up with languages (and notations) to model the set of relevant concerns of an enterprise such as intent or goals, operational processes, organizational structure, services, etc. These are enterprise architecture-ish models but machine manipulable so as to support automated what-if, if-what, change impact analyses among others. It will also be possible to establish functional and extra-functional properties of an enterprise in qualitative and/or quantitative terms. We will also investigate possibility of arriving at a transformation plan from as-is state of enterprise to the desired to-be state. We already have early results in specifying and analyzing enterprise models using ontological representations [16]. Work on What-if, if-what scenario analysis [17] and scenario playing [18] is currently submitted.

Enterprise Simulation. At present, techniques and technology exist to simulate an enterprise architecture model or a specific set of concerns individually and independently. However, simulating an enterprise would need simulation of all its models in concert which is not possible today. System dynamical models are characterized by a small set of primitives and powerful simulation machinery [19]. This work-package constitutes evaluation of system dynamical models for enterprise modeling, developing simulation machinery for EA models that can be specified as above, and investigating if a link can be established between EA models and system dynamical models. Objective here is to advance state of art in simulation and applications/applicability of simulation techniques.

IT Plant Specification. This work-package constitutes specifying various concerns of IT plant so as to generate a configurable extensible platform implementation using it and to help monitor, evolve and adapt IT plant under human supervision. At present it is possible to specify the various concerns of a business application e.g. user interface, data and data access, online and batch functionality, reports etc in a model form from where an efficient implementation can be effectively and efficiently generated. It is also possible to specify interactions between applications as an orchestration or choreography. Early advance has taken place as regards design-time and run-time configuration of an application. However, little work is reported on application architecture to support a priori unknowable extensibility. Business Process Platforms (BPP) providing a set of business processes and their automation through a set of services *out of the box* was a buzzword in 2005 which has remained unrealized. Early advance in adaptation architectures is limited to individual applications. The adaptation concept needs to be extended to other constituents of IT plant such as business processes, batch programs etc individually and to the whole IT plant collectively. Objectives of this work-package are to come up with i) an implementation architecture for BPP with additional requirement of support for easy configuration and a priori unknowable extensibility of the entire BPP, ii) the implementation machinery to realize an IT plant, and iii) adaptation architecture for individual components as well as the whole of IT plant.

IT Plant Contract Specification. It should be possible to specify a set of concerns such as functional, operational, legal, monetary etc, and thus, in essence, forms a contract between IT plant provider and consumer. From IT plant consumer's perspective, it should be possible to specify the desired IT needs in terms of all the relevant aspects such as functionality, data, user experience, extra-functional characteristics [15] etc.

IT Plant Testing. At present it is possible to specify application behavior at a higher level from which test cases and test data for system testing with coverage related assurance can be generated. Early advance is underway as regards testing of a product line depicting a set of applications having high commonality and well-defined variability. Automation harnesses for regression testing have been around for years but incremental i.e. change-specific testing is still a problem. Objective of this work-package is to extend these concepts to cover the whole of IT plant i.e. a set of applications, a set of business processes, a set of batch programs, a set of interfacing channels etc. Another, and probably more important, problem is to establish testability of the IT plant. Another aspect of IT plant testing is in relation to satisfaction of contract.

IT Plant Deployment. It is highly unlikely that a consumer of IT plant won't already be using IT systems. Typically, IT plant will replace some of the IT systems in use. As systems being replaced (say A) might be interacting with the systems not being replaced (say B), B systems need to be modified as regards their dependence on A systems in terms of service calls, data in/out etc and the IT plant needs to be extended to cater to interfacing needs of A systems. Availability demand dictates these modifications need to be undertaken conforming to a partial order and in batches. Identifying both may need analysis of implementation of existing systems and/or execution logs. Having identified the partial order, B systems need to be suitably updated, IT plant suitably extended, and A systems decommissioned. These activities should be automated to the extent possible.

6 Related Work

Servicing of outsourced IT systems in an innovative way as we have suggested is novel in the sense that such business model and analysis and operational views of this business model have not been suggested/researched previously to best of our knowledge. We therefore take review of key works that we refer to, including online resources since the sentiment that something more than the prevalent cost arbitrage model is required for servicing outsourcing has started surfacing only recently.

What should enterprises outsource and what they should not and the benefits and risks involved was discussed in [9]. It was found that labor cost reduction is the most ranked benefit whereas political and legal issues including property rights and contracts were counted as most ranked risks. In contrast to this study which was carried out in 2005, recent sentiment suggests that enterprises are looking beyond cost benefits, specifically to value generation [10, 11] something that analysis world view of IT plant may be able to provide to some extent [12].

How such value-oriented relationship might exist between enterprises and SPs is being researched as described in [1, 13]. Scoping and scaling of work to be performed by SP needs to be well defined [13] and can take form of strategic relations, co-sourcing alliance, or transaction exchange [1]. It is suggested in [1] that the relationships stated above evolve and that enterprise must transition from strategic relationship to transaction exchange in a step-by-step manner. This remains to be tested further. We on the other hand believe that once the suggested work-packages have been realized, it might be possible for us to get into any relationship with an enterprise aided by precise specification of IT plant and enterprise, and analysis and operationalization abilities.

7 Conclusion

It is quite evident in recent times that focus on outsourcing enterprise IT plants has shifted from traditional concerns like cost reduction, access to skilled workforce, continuous operations, agility and so on, to value generation. We have shown in what way such value-oriented IT plant may be realized. From SP's perspective, in a given vertical, it is quite possible to service IT plants of multiple enterprises as there ought to be much that is common and with few variations. We already have some early results with regards to various work-packages that we have specified. It is our belief that future IT plant outsourcing will be mainly value-driven and we have indicated how we and other SP's may benefit from a new business model and how enterprises might benefit with analysis and operational world views of their IT plants.

References

1. Ross, J.W., Beath, C.M.: Sustainable IT Outsourcing Success: Let Enterprise Architecture Be Your Guide. MIS Quarterly Executive 5 (2006)
2. Kim, B.S.B.S.: Systematic Prioritization Of Consideration. In: Making Offshore Software Development Outsourcing Decisions. Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science (2005)
3. Kulkarni, V., Venkatesh, R., Reddy, S.: Generating enterprise applications from models. In: Bruel, J.-M., Bellahsène, Z. (eds.) OOIS 2002 Workshops. LNCS, vol. 2426, pp. 270–279. Springer, Heidelberg (2002)
4. Kulkarni, V., Reddy, S.: Model-Driven Development of Enterprise Applications. In: Jardim Nunes, N., Selic, B., Rodrigues da Silva, A., Toval Alvarez, A. (eds.) UML Satellite Activities 2004. LNCS, vol. 3297, pp. 118–128. Springer, Heidelberg (2005)
5. Kulkarni, V., Reddy, S.: Introducing MDA In a Large IT Consultancy Organization. In: APSEC, pp. 419–426. IEEE Computer Society (2006)

6. Kulkarni, V., Reddy, S., Rajbhoj, A.: Scaling Up Model Driven Engineering – Experience and Lessons Learnt. In: Petriu, D.C., Rouquette, N., Haugen, Ø. (eds.) MODELS 2010, Part II. LNCS, vol. 6395, pp. 331–345. Springer, Heidelberg (2010)
7. Sunkle, S., Kulkarni, V.: Cost Estimation for Model-Driven Engineering. In: France, R.B., Kazmeier, J., Breu, R., Atkinson, C. (eds.) MODELS 2012. LNCS, vol. 7590, pp. 659–675. Springer, Heidelberg (2012)
8. Kulkarni, V., Barat, S., Roychoudhury, S.: Towards Business Application Product Lines. In: France, R.B., Kazmeier, J., Breu, R., Atkinson, C. (eds.) MODELS 2012. LNCS, vol. 7590, pp. 285–301. Springer, Heidelberg (2012)
9. Djavanshir, G.R.: Surveying the Risks and Benefits of IT Outsourcing. IT Professional 7, 32–37 (2005)
10. Overby, S.: Decline in IT Outsourcing Market Challenges Service Providers (2013), `http://www.cio.com/article/727586/Decline_in_IT_Outsourcing_Market_Challenges_Service_Providers`
11. Bahli, B.: Enabling Innovation In Information Technology Outsourcing: An Empirical Study. In: 23rd Australasian Conference on Information Systems, Geelong, Victoria, pp. 1–8 (2012)
12. Overby, S.: IT Outsourcing Customers Get What They Pay for, Not What They Want (2013), `http://www.cio.com/article/729337/IT_Outsourcing_Customers_Get_What_They_Pay_for_Not_What_They_Want`
13. Overby, S.: 5 Ways to Build Better IT Outsourcing Relationships (2013), `http://www.cio.com/article/720000/5_Ways_to_Build_Better_IT_Outsourcing_Relationships`
14. Markus, M.L., Tanis, C.: The Enterprise Systems Experience-from Adoption To Success. In: Framing the Domains of IT Management: Projecting the Future Through the Past, pp. 173–207. Pinnaflex Education Resources, Inc. (2000)
15. Lacity, M.C., Rottman, J.W.: Offshore Outsourcing of IT Work: Client And Supplier Perspectives Palgrave Macmillan, Basingstoke, England, New York (2008)
16. Sunkle, S., Kulkarni, V., Roychoudhury, S.: Analyzable Enterprise Models Using Ontology. In: To be included in Proc. CAiSE Forum 2013 (2013)
17. Sunkle, S., Kulkarni, V., Roychoudhury, S.: Intentional Modeling for Problem Solving in Enterprise Architecture. In: ICEIS (Working paper- submitted for review)
18. Sunkle, S., Roychoudhury, S., Kulkarni, V.: Using Intentional and System Dynamics Modeling to Address WHYs in Enterprise Architecture. In: ICSOFT-EA (Working paper-submitted for review)
19. Sterman, J.D.: Business Dynamics: Systems Thinking and Modeling for a Complex World. Irwin/McGraw-Hill (2000)

Characteristics of Knowledge and Barriers towards Innovation and Improvement in Collaborative Manufacturing Process Chains

Benjamin Knoke, Thorsten Wuest, and Klaus-Dieter Thoben

BIBA – Bremer Institut für Produktion und Logistik GmbH,
Hochschulring 20, 28359 Bremen, Germany
{kno,wue,tho}@biba.uni-bremen.de

Abstract. The characteristics of knowledge relevant to initiate innovation projects or improve existing manufacturing processes have been identified against the background of an organisation participating in a collaborative manufacturing process chain. Those innovation or improvement processes rely on differing types of knowledge, which must be combined in order to determine the current state, the desired state, and the methodology to get there. In a collaborative environment, all of these knowledge types can be fragmented and stored within certain partitions resembled by people, or organisational units. Within this paper, two different organisational units, management and employees of a focal organization or those of related partners have been selected to illustrate the problem. By adapting the Johari window to map knowledge exchange, this paper identifies the characteristics of intra- and inter-organisational barriers.

Keywords: Knowledge Management, partitioned knowledge, Johari Window, operations improvement, collaborative innovation, manufacturing process chain, collaborative manufacturing.

1 Introduction

A manufacturing process chain can be defined as specific manifestation of a series of manufacturing processes that focuses on the physical transformation of tangible goods [1]. In industry, these processes are usually distributed between multiple people, highlighting information and knowledge exchange as important research areas [2]. The significance of such communication can be estimated according to a study conducted by [3], which calculated costs of $611bn per year caused by poorly targeted mailings and staff overheads in the US,. [3] also stated that organizations typically overestimate the quality of their data and at the same time underestimate the cost of potential errors. The impact of communication amplifies with rising product complexity: while a complex product requires knowledge and skills from different fields, the product development and the resulting manufacturing process chain are often designed by teams, composed of experts from collaborating organisations [4]. To maintain and improve such collaboration, a successful exchange of knowledge is mandatory.

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 264–273, 2013.

This paper aims to shed light on the characteristics of knowledge that is exchanged within manufacturing process chains, and the barriers involved as a problem description. To elaborate on a systematic problem analysis, this paper builds upon an adaptation of the Johari window for intra- and inter-organizational knowledge exchange that has been conducted by previous research in the manufacturing domain.

2 Characteristics of Relevant Knowledge to Improve Collaborative Manufacturing Process Chains

Knowledge Management is the systematic and explicit control of knowledge based activities, programs and governance within the enterprise with the goal to make effective and profitable use of the intellectual capital [5]. The Knowledge Management research field is a very broad one and there are various research areas involved, from social science over psychology and business to engineering and many more [6]. [7] emphasize that Knowledge Management does not only imply successful utilization of knowledge but also creation, identification, allocation, development, usage, conservation and sharing [8]. Knowledge is a key resource for enterprises [9] [10] and sharing knowledge is crucial for every modern manufacturing company especially when working in a collaborative environment [11]. Sharing knowledge is always a challenge as it contains a context dimension in comparison to e.g. information [11]. Knowledge can be available implicitly (e.g. in the head of employees) or in explicit form (e.g. documentation) [12], with some researchers arguing that even explicit knowledge is partly implicit as the person acquiring the knowledge automatically interprets it [13].

The pioneers in the field of Knowledge Management, [14] created the well-known model of the "knowledge spiral", an illustration of the knowledge creating process focusing on transforming implicit to explicit knowledge. The complexity of such externalization is however dependant on the characteristics of the intended knowledge transfers. While some knowledge can be easily externalized, e.g. creating a technical documentation of a simple product; more advanced knowledge may require complex training mechanisms, e.g. how to operate a milling machine. Such difficulties in externalizing complex knowledge lead, in combination with non-ideal communication, to the fact that each person possesses an individual set of knowledge [15]. Therefore, whenever multiple persons collaborate by sharing knowledge, they face the problem of partitioned knowledge. Such knowledge partitioning is the theoretical notion that independent non-overlapping parcels contain individual sets of knowledge, which may result in people making contradictory decisions for identical problems under different circumstances [16].

The New Product Development (NPD) process can be described as a knowledge-intense activity [17], [18]. Each participant of the NPD process can rely on an individual set of relevant knowledge. Such knowledge can be considered relevant, if it has a positive impact on the collaborative endeavour. Approaches to characterize knowledge relevant for NPD have been made by [19] in distinguishing between two types of memory systems: declarative knowledge (what?), which contains all knowledge about facts and events; and procedural knowledge (how?), which contains methodologies and provides the ability to act and realize even complex tasks, such as NPD.

The declarative element comprises a broad field that has been narrowed down by consecutive research. [20], [21], [22] identify three critical types of knowledge: *domain-specific* knowledge, *general* knowledge, and *procedural* knowledge. Although, these three types partly influence each other, they are characterized as independent categories [23]:

- **Domain-specific knowledge** is a type of knowledge about the form or function of an individual object or class of objects [23] and has been gained through previous results, in similar or related activities [19], [20]. Either stored tacit or explicitly, it can be used in future NPD projects.
- **General knowledge** is made through everyday experiences and general education [20]. It comprises knowledge that connects domain-specific knowledge about what is happening outside of the organisation
- **Procedural knowledge** is the knowledge about the development process itself [24], and is therefore closely linked to methodological competence. It can be defined as "the knowledge of what to do next" [22]. Many organisations store procedural knowledge in formal routines or process instructions [25].

Contrary to the declarative/procedural knowledge approach, these three types do not cover all types of knowledge that NPD could possibly benefit from. [21] states that other types, such as knowledge about production processes should be considered against this background as well.

The above characteristics apply as well to knowledge that is relevant to the task of improving a collaborative manufacturing process. These types however, need to be extended to cover knowledge about the current as-is state, which is crucial for any improvement activity.

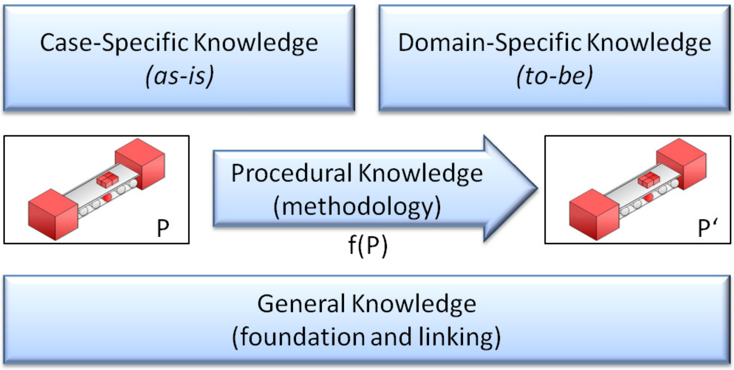


Fig. 1. Knowledge characteristics for improvement of manufacturing process chains

The results, as shown in **Fig. 1**, are four types relevant for improvement or innovation projects based on collaborative manufacturing chains:

- **Case-specific-knowledge** about the current as-is state of the targeted process chain must be collected to successfully derive or implement any improvement measures.
- **Domain-specific knowledge** is crucial to determine opportunities for improvement and to check the feasibility of options.
- **Procedural knowledge** comprises the methodological knowledge and describes how other types of knowledge need to be applied.
- **General knowledge** is the overall foundation of the improvement activities. It is accessed to select the methodology and domain-specific knowledge depending on the current improvement activity.

These types of knowledge apply to process chain improvement in single organisations as well as to organisations in a collaborative environment. The latter however, faces the problem of knowledge fragmentation and resulting boundaries, which are described in more detail in the following section.

3 Barriers between Knowledge Partitions in Collaborative Manufacturing Process Chains

From a knowledge management perspective, a network or even a single organization cannot be considered a collective. Instead they consist of individuals with individual partitions of knowledge. Whenever such individuals collaborate by sharing their knowledge, they communicate to connect their knowledge partitions and to overcome barriers between them. This section builds upon a knowledge management concept of knowledge being fragmented and stored in several partitions throughout a collaborative manufacturing chain. The concept has been previously developed by [26] and features an adoption of the Johari window for inter-organisational use.

3.1 The Johari Window for Inter-organisational Knowledge Management

The Johari window itself is an analytic tool that has been developed in 1955 by Joseph Luft and Harry Ingham [27]. As shown in Fig. 1, the Johari window is a 2x2 matrix that consists of four panels, which contain personal attributes. These personal attributes are sorted by their awareness to the referred person and to the group, thus resulting in four categorizing areas:

- **The Arena**, contains overall attributes that are known to the person, as well as to the group
- **The Blind Spot**, is comprised of knowledge that is available to the group, but not to the referred person
- **The Façade**, which holds attributes that are only known to the referred person and unknown to the group
- **The Unknown**, which contains attributes not known to anyone

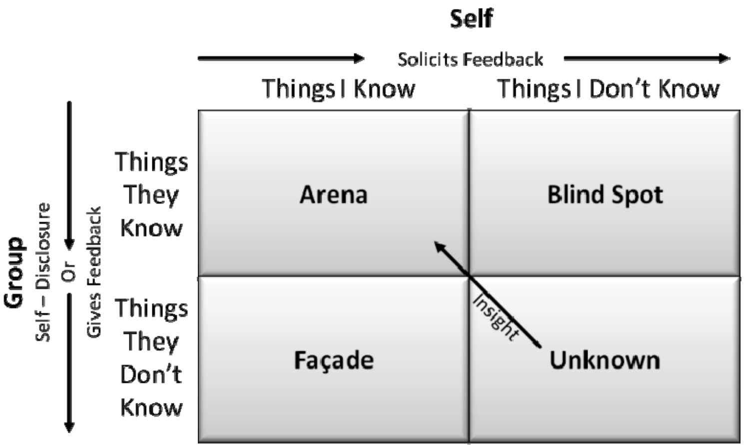


Fig. 2. The Johari Window

Regardless of its age, the Johari Window is still a popular tool to study feedback and exposure processes in various sectors [28], [29], [30], [31]. Its success is based upon its simplicity and extensibility. By adjusting the size of its panels according to their content, the Johari Window can be used to simulate and visualize dynamic feedback [32], [33].

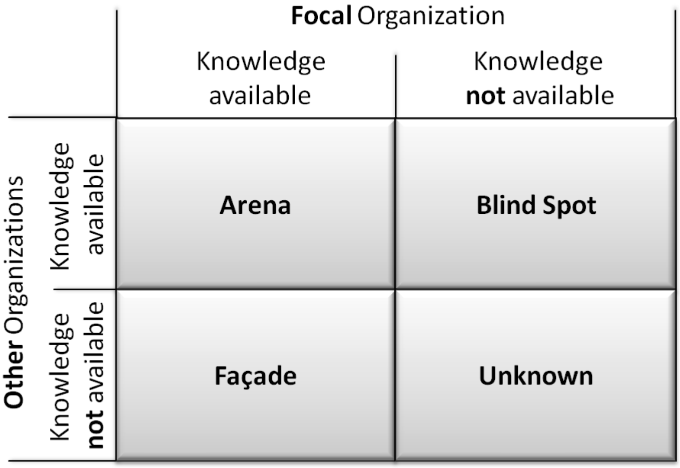


Fig. 3. The Johari-Window for inter-organizational Knowledge Management

The Johari window has been adopted by [26] to map the knowledge of a focal organisation. This results in a 2x2 matrix to sort knowledge that is known/not known to this focal organisation or others, as shown in **Fig. 3**.

3.2 Characteristics of Intra- and Inter-organisational Barriers for Knowledge Exchange

This section elaborates on the characteristics of barriers for knowledge about improvement potential within an organisation that acts in a collaborative manufacturing process chain.

Based on the assumption that measures to improve an organisation's manufacturing are taken by its management, but practical knowledge is gathered by its employees, an exchange of such knowledge leads to the best use of the organisation's potential. If employees and managers are defined as partitions that store certain fragments of knowledge, those partitions are separated by intra-organisational barriers, which are crossed by communication.

With the aim of process improvement in mind, an organisation that participates in a collaborative manufacturing process chain connects with other organisations and exchanges case-specific knowledge. From their differing knowledge base and point of view, those connected organisations might identify potential improvements for the focal organisation's manufacturing processes. To gather such beneficial knowledge about improvement potential, the focal organisation needs to overcome an inter-organisational barrier.

Both barriers are visualized in **Fig. 4**. Organisations should generally consider to access the knowledge of their employees with the highest priority, followed by the management of other organisations. Reaching for knowledge of employees affiliated to other organisations should be conducted with highest precaution, as reactions on such activities depend on the level of trust and type of the collaboration.

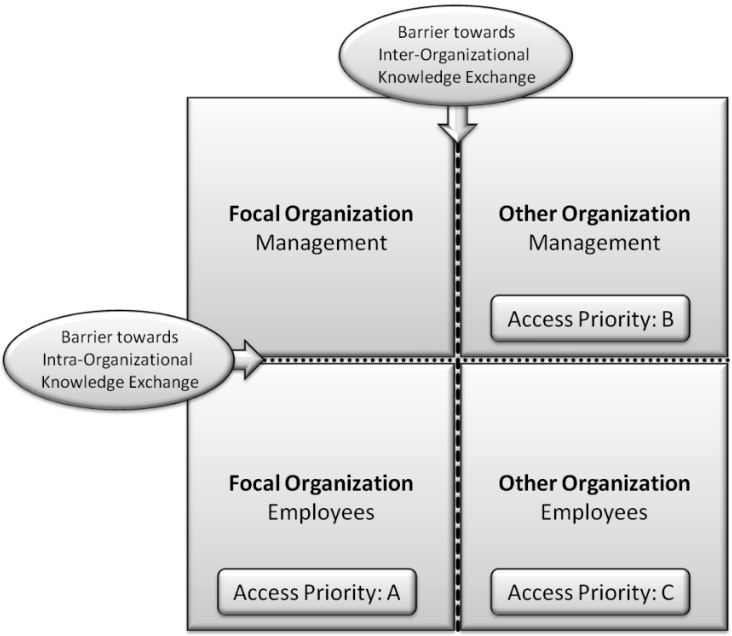


Fig. 4. Barriers for Intra- and Inter-Organizational Knowledge Exchange [26]

The characteristics of barriers towards communication in general are a diffuse, but strongly researched area. According to [34] and [35], significant barriers arise from personal differences in various manifestations, such as different culture, age, language, or personal disposition and character.

Other categories can be identified through encouragement, the organizational support, and the committed resources [36]. Along with the most relevant personal differences, the barriers can be characterized, as shown in **Fig. 5**.

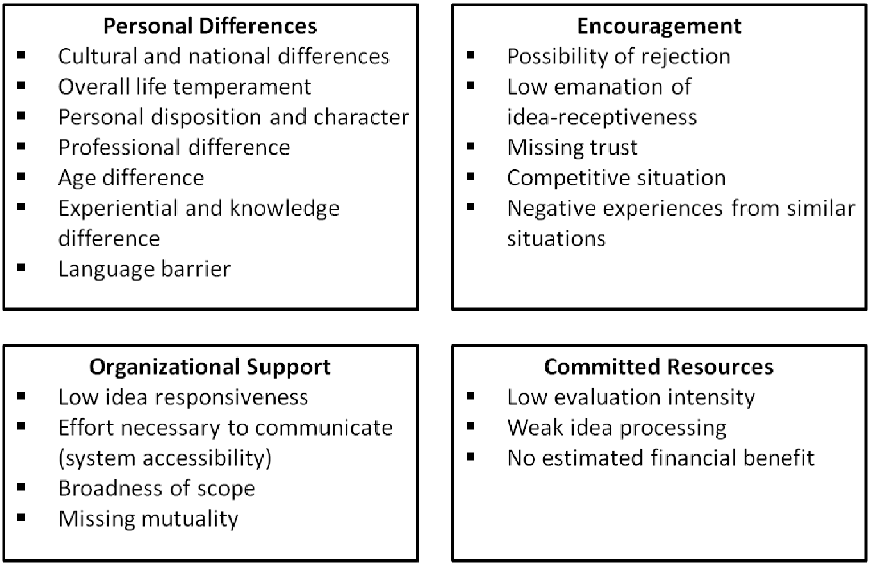


Fig. 5. Barriers for intra- and inter-organisational knowledge exchange (excerpt)

The barrier of personal differences not necessarily requires knowledge about the corresponding person; instead it can be based on assumptions or prejudices, but is always based on at least estimated differences in the characteristics of the respective persons. The four described categories are not independent from each other, but instead impact on the encouragement of a person, which can be considered the precondition for any successful attempt of knowledge transfer between organizations, organization units, or individuals.

4 Limitation and Outlook

The characteristics of knowledge relevant to initiate innovation projects or improve existing manufacturing have been identified against the background of an organisation participating in a collaborative manufacturing process chain. Those innovation or improvement processes rely on a general knowledge to identify the procedural knowledge that must be combined with case-specific knowledge to achieve a targeted state, which is determined by domain-specific knowledge.

In a collaborative environment, knowledge can be fragmented and stored within certain partitions resembled by people, or organisational units. The organisational units identified within this paper are management and employees of a focal organisation, or those of related partners. With adoption of the Johari window to map knowledge exchange, this paper identifies the characteristics of intra- and inter-organisational barriers.

The research is limited in the form that management and employees have been looked at as solid entities, in fact knowledge could be available to people within the management of an organisation, but not to the responsible process owner. This results in an additional intra-fractional barrier that has not been described to this point.

Subsequent research will focus on analysing those previously described barriers more thoroughly, concepts to overcome those barriers and cover the relation of Open Innovation approaches and knowledge diffusion in collaborative environments. Additional potential arises from an impact analysis of material flows on knowledge exchange, possibly conducted by intelligent products, while they flow through such process chains; the potential of intelligent cargo systems; or an innovative logistics concept focussing on transferring not only the product, but also knowledge within a collaborative environment.

Acknowledgment. This work has been funded by the European Commission through ICT Project BIVEE: Business Innovation and Virtual Enterprise Environment (grant agreement: 285746). The authors wish to acknowledge the Commission for their support. We also wish to acknowledge our gratitude and appreciation to all the BIVEE project partners for their contribution during the development of various ideas and concepts presented in this paper.

References

1. Thomas, O., Walter, P., Loos, P.: Design and usage of an engineering methodology for product-service. Journal of Design Research 7(2), 177–195 (2008)
2. Oztemela, E., Tekez, E.K.: Integrating manufacturing systems through knowledge exchange protocols within an agent-based Knowledge Network. Robotics and Computer-Integrated Manufacturing 25(1), 235–245 (2009)
3. Marsh, R.: Drowning in dirty data? It's time to sink or swim: A four-stage methodology for total data quality management. Database Marketing & Customer Strategy Management 12(2), 105–112 (2005)
4. Kleinsmann, M., Buijs, J., Valkenburg, R.: Understanding the complexity of knowledge integration in collaborative new product development teams: A case study. Journal of Engineering and Technology Management 27, 20–32 (2010)
5. Wiig, K.M.: Perspectives on introducing enterprise knowledge management. In: Proceeding of the CEUR Workshop, vol. 13, RWTH Aachen, Aachen (1998)
6. Wuest, T., Thoben, K.-D.: Exploitation of Material Property Potentials to Reduce Rare Raw Material Waste - A Product State Based Concept for Manufacturing Process Improvement. Journal of Mining World Express (MWE) 1(1), 13–20 (2012)
7. Davenport, T.H., De Long, D.W., Beers, M.C.: Successful knowledge management projects. Sloan Management Review 2(39), 43–57 (1998)

8. Probst, G.J.B.: Practical Knowledge Management: A Model That Works. In: Little, A.D. (ed.) Prism, vol. 2, pp. 17–29 (1998)
9. Gehle, M.: International Knowledge Management. For increase of flexibility and economic power of knowledge focused enterprises (original German title: Internationales Wissensmanagement. Zur Steigerung der Flexibilität und Schlagkraft wissensintensiver Unternehmen). Deutscher Universitäts-Verlag, Wiesbaden (2006)
10. Reichwald, R., Piller, F.: Interactive value creation. Open innovation, individualization, and new forms of work distribution (original German title: Interaktive Wertschöpfung. Open Innovation, Individualisierung und neue Formen der Arbeitsteilung). Gabler Verlag, Wiesbaden (2009)
11. Swart, J., Kinnie, N.: Sharing knowledge in knowledge-intensive firms. Human Resource Management Journal,vol 2(13), 60–75 (2003)
12. Haun, M.: Handbook Knowledge Management. Basics and Realization, Systems and Praxis Examples (original German title: Handbuch Wissensmanagement. Grundlagen und Umsetzung, Systeme und Praxisbeispiele). Springer Verlag, Berlin (2002)
13. North, K., Güldenberg, S.: Productive knowledge work(er) – answers to the 21st century management challenges (original German title: Produktive Wissensarbeit(er) – Antworten auf die Managementherausforderungen des 21. Jahrhunderts). Gabler Verlag, Wiesbaden (2008)
14. Probst, G., Raub, S., Rombhardt, K.: Managing Knowledge – How Enterprises make optimal benefit of their most valuable resource (original German title: Wissen managen – Wie Unternehmen ihre wertvollste Ressource optimal nutzen). Gabler Verlag, Wiesbaden (2006)
15. Bhattacharya, M., Jorgensen, L.: Minimizing Plagiarism by Redesigning the Learning Environment and Assessment. In: Student Plagiarism in an Online World: Problems and Solutions, Tim Roberts, Information Science Reference, Hershey, New York (2008)
16. Lewandowsky, S., Kirsner, K.: Expert knowledge is not always integrated: A case of cognitive partition. Memory & Cognition 28, 295–305 (2000)
17. Brockman, B.K., Morgan, R.M.: The role of existing knowledge in new product innovativeness and performance. Decision Sciences 34(2), 385–419 (2003)
18. Lawson, B., Petersen, K.J., Cousins, P.D., Handfield, R.B.: Knowledge sharing in interorganizational product development teams: The effect of formal and informal socialization mechanisms. Journal of Product Innovation Management 26, 156–172 (2009)
19. Berge, T., ten, H.R.: van: Procedural and Declarative Knowledge. An Evolutionary Perspective. Theory & Psychology 9(5), 605–624 (1999)
20. Court, A.W.: The relationship between information and personal knowledge in new product development. International Journal of Information Management 17(2), 123–138 (1997)
21. Ramesh, B., Tiwana, A.: Supporting collaborative process knowledge management in new product development teams. Decision Support Systems 27, 213–235 (1999)
22. Frishammar, J., Lichtenthaler, U., Rundquist, J.: Identifying Technology Commercialization Opportunities: The Importance of Integrating Product Development Knowledge. Journal of Product Innovation Management 29(4), 573–589 (2012)
23. Ullman, D.G.: The mechanical design process. McGraw-Hill, Boston (2010)
24. Kyriakopoulos, K., de Ruyter, K.: Knowledge stocks and information flows in new product development. Journal of Management Studies 41(8), 1469–1498 (2004)
25. Olivera, F.: Memory systems in organizations: An empirical investigation of mechanisms for knowledge collection, storage, and access. Journal of Management Studies 37(6), 811–832 (2000)

26. Knoke, B., Wuest, T., Thoben, K.-D.: Fragmented Knowledge in Collaborative Manufacturing Process Chains. Accepted for Publication at the, International Conference on Collaboration Technologies and Systems, CTS 2013, San Diego, California (2013)
27. Luft, J.: The Johari Window: A Graphic Model of Awareness in Relations. National Press Books, Palo Alto (1970)
28. Chang, W.-W., Chen, C.-H.L., Huang, Y.-F., Yuan, Y.-H.: Exploring the Unknown: International Service and Individual Transformation. Adult Education Quarterly 62(3), 230 (2012)
29. Roland, D., Matheson, D.: New Theory from an old Technique: the Rolma Matrices. The Clinical Teacher 9, 143–147 (2012)
30. Barratt, W.: Social Class and the Extracurriculum. Journal of College & Character 13(3) (2012)
31. Nair, S.K., Naik, N.S.: The Johari Window Profile of Executives of a Public Sector Undertaking. Management and Labour Studies 35(2), 137–148 (2012)
32. Nicholls, J.: The Ti-Mandi window: a time-management tool for managers. Industrial and Commercial Training 33(3), 104–109 (2001)
33. Shirland, L.E.: Dynamics of the Johari Window: A Simulation. In: Winter Simulation Conference, pp. 396–401 (1977)
34. Nijkamp, P., Rietveld, P., Salomon, I.: Barriers in spatial interactions and communications - A conceptual exploration. The Annals of Regional Science 24, 237–252 (1990)
35. Klimova, B.F., Semradova, I.: Barriers to communication. Procedia - Social and Behavioral Sciences 31, 207–211 (2012)
36. Van Dijk, C., Van den Ende, J.: Suggestion systems: transferring employee creativity into practicable ideas. R&D Management 32(5), 387–395 (2002)

A Logic-Based Formalization of KPIs for Virtual Enterprises*

Claudia Diamantini, Domenico Potena, and Emanuele Storti

Dipartimento di Ingegneria dell'Informazione,
Università Politecnica delle Marche - Via Brecce Bianche, 60131 Ancona, Italy
{c.diamantini,d.potena,e.storti}@univpm.it

Abstract. Open innovation is gaining increasing interest as a model to foster innovation through collaboration and knowledge sharing among organizations, especially in the context of Virtual Enterprises (VE). One of the main issues to overcome in such distributed settings is the integration of heterogeneous data, and the need to evaluate common Key Performance Indicators (KPI) capable to measure overall performances of the VE. In this paper we propose a conceptualization of KPIs into an ontology, to provide a common vocabulary to semantically annotate data belonging to different organizations. KPIs are described in terms of dimensions and a mathematical formula. In order to support reasoning services over KPIs formulas we refer to a logic-based formalization in Prolog, where formulas are translated as facts, and several predicates are included to support both mathematical functionalities for formula manipulation and higher-level functions especially suited for VE setup.

Keywords: KPI ontology, logic-based formalization of KPIs, logic reasoning, open innovation, virtual enterprises.

1 Introduction

Collaboration, communication and interaction are key elements to open paradigms to innovation and especially to Virtual Enterprises (VE), which are characterized by heterogeneous groups of partners with different policies, skills, know-how and way of doing business. In this open context, the BIVEE project[1] is purposed to develop an ICT framework capable to address the need of knowledge integration and sharing for innovation projects. In BIVEE, to overcome interoperability issues, a semantics-based infrastructure, namely the Production and Innovation Knowledge Repository (PIKR) [1], serves as a knowledge gateway. By enabling a unified access to information and by exploiting a set of reasoning functionalities, it is capable to grant a smart and easier access to resources and ultimately to enhance support for decision making.

* This work has been partly funded by the European Commission through the ICT Project BIVEE: Business Innovation and Virtual Enterprise Environment (No. FoF-ICT-2011.7.3-285746).

[1] http://www.bivee.eu

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 274–285, 2013.

In order to compare performances of the participating organizations, to measure overall performances of the VE and ultimately to enable managers to define information requirements and goals, it becomes crucial to evaluate heterogeneous Key Performance Indicators (KPI) about innovation and production. This problem can be considered a particular instance of the more general integration of federated Data Warehouses (DWH), in which each organization provides a DWH with its own dimensional structure and fact table, and KPIs are the measures for this last. However, besides the typologies of heterogeneity usually addressed in the DWH Literature (i.e., dimensional heterogeneities, see [2] for a survey), in distributed and collaborative environment a further source of heterogeneity is given by the semantics that each organization assumes for the indicators, in terms of both conceptual meaning and mathematical structure of the corresponding formula [3]. For instance, let us consider two collaborating enterprises A and B, providing data about production costs measured through different indicators, namely I_x (for A) and I_y, I_z (for B). Given that there are not indicators in common, integration at VE level would not be feasible. However, if the structure of I_x formula is know and equal to a combination of the other two, e.g. $I_y + I_z$, then it is possible to compute I_x also for B and compare the two results.

In the context of the PIKR, this work is aimed to present the semantic model for the definition and manipulation of heterogeneous KPIs belonging to collaborating organizations that take part to a VE. The approach is based on the conceptualization of KPIs, including their properties and the structure of their formulas, through an ontological representation that serves as a common reference language for the VE, allowing the annotation of enterprise data by means of a shared terminology. Then, in order to support reasoning functionalities over KPIs, we refer to a logic-based formalization in predicate logic of KPI formulas and mathematical functions implementing axioms, together with operators for equation solving and formula manipulation. By following this approach, indicators referring to the same concept but having a different name or mathematical structure can be reconciled, and new indicators can be defined in the VE, if implicitly computable starting from the provided data. The reasoning services introduced herein are targeted to support the VE setup phase, and include KPI elicitation, the analysis of dependencies among indicators in terms of common components, evaluation of semantical equivalence between formulas, check of consistency and correctness. The management of annotation of enterprise data, as well as the definition of links between business concepts and KPIs are outside the purpose of this work, as they are addressed by specific modules within the BIVEE framework.

The rest of this work is organized as follows: next Section proposes a brief overview of the main solutions and models available in the Literature for KPI modeling and management. In Section 3 the conceptualization of KPIs into an ontology is provided, while Section 4 is devoted to present a logic-based formalization of KPI formulas in predicate logic, together with the reasoning services. A case study in an example scenario is introduced in Section 5, while Section 6 provides some final remarks and discusses future extensions.

2 Related Work

Only few work in the Literature have focused to models and methodologies for monitoring efficiency and effectiveness of innovation activities and for innovation project planning, especially in cooperative environments. Many studies propose innovation indicators at macro-level, suitable to analyse innovation performances of countries or market segments (see e.g., [4]) but largely unapt to monitor and improve internal processes and organization structures. In [5] a review of the Literature pertaining to the measurement of innovation management at the firm's level is presented, reporting also "an absence of measures well aligned to the activities of the innovation process". A framework and a set of 26 KPIs for the measurement of innovation availability, efficiency and effectiveness has been proposed in [6]. In such a work, innovation in a firm is seen as a whole, ignoring the process that created it. A framework to semantically define Performance Indicators related to processes is introduced in [7]. However, our focus is not limited to KPIs about processes, as we address also indicators about other resources and perform reasoning on them. To the best of our knowledge, there is a lack of work about reconciliation of heterogeneous KPIs used by different enterprises in a collaborative setting, which requires a peculiar care in the management of KPI integration.

For what concerns a systematization of KPI for enterprise environments, there is a plethora of Performance Indicators definitions and/or Glossaries provided by researchers and research groups, international and national public bodies (e.g., [4]), and in reference models like the Supply-Chain Reference Model (SCOR) [8], the Value Chain Reference Model (VRM) [9] and Six-Sigma [10] (see also [1] for a more comprehensive list of performance measurement approaches and KPI description). All such reference models provide quite diverse indicators, depending on the goal and the domain of interest. However, in order to design the ontology described in Section 3, namely KPIOnto, we need to abstract over their general characteristics so that any indicator can be represented. In this sense, almost all proposals are homogeneous in the kinds of information provided, differing on the degree of precision and formality of the description. For the design of the ontology we focus on the description of indicators in VRM. This choice has three main reasons: (1) the VRM model explicitly addresses networked enterprises, (2) the model provides an accurate description of indicators, and (3) the BIVEE project mainly refers to VRM model to develop its business innovation reference framework [1].

A first proposal of the ontology and the approach described in this work is available in [3], in which a semantic representation of KPIs for health care sector is presented, together with a set of reasoning functionalities aimed at requirements elicitation and DWH design, in particular for data marts materialization, consolidation and integration.

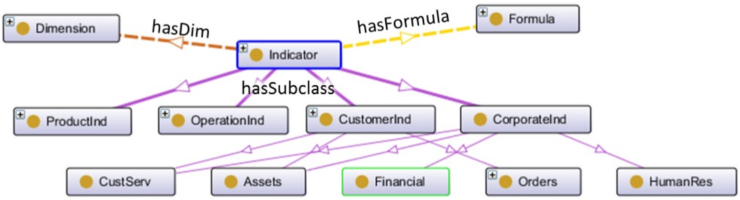


Fig. 1. KPIONTO: main classes

3 KPIOnto: An Ontology of Key Performance Indicators

Following the VRM model, in KPIOnto a taxonomy of KPIs is introduced, based on the Enterprise area of intervention, e.g., Corporate, Customer, Operation or Product. However, multiple-categorization is allowed by the ontology, thus supporting more efficient indexing and searching functionalities. The main categories of the KPIOnto that are relevant in the context of this work are Indicator, Dimension and Formula. Figure 1 shows these classes and relationships among them. The *Indicator* is the pivotal class of the KPIOnto, and its instances (i.e., indicators) describe the metrics enabling performance monitoring. Properties of an indicator are: name, identifier, acronym, definition (i.e., a plain text giving a detailed description of the indicator meaning and its use) and Unit of measurement (i.e., both the symbol and the description of the unit of measurement chosen for the indicator. These properties are given by referring to the Measurement Units Ontology (MUO) [11]).

A dimension is the coordinate/perspective to which the metric refers. For instance, it is useful to analyse deliveries along dimensions like the delivery date, the delivered product, the means of transportation, and so forth. Following the multidimensional model, a dimension is usually structured into a hierarchy of levels, where each level represents a different way of grouping elements of the dimension [12]. For instance, it can be useful to group means of transportation by transport companies, and days by weeks and years. Each level is instantiated in a set of elements known as members of the level, e.g. the company "ACME", the weeks "3rd-2012" and "42nd-2011". In order to describe dimensions in KPIOnto, our approach is similar to the one proposed in [13]. In particular, the top class is *Dimension*, which is composed of a set of subclasses, one for each specific dimension, like Organization Dimension, Time Dimension, and so forth. Levels are represented as disjoint primitive subclasses of the dimension (or the set of dimensions) they belong to, each member is an instance of these classes, and the dimension's hierarchy is represented by a part-of relationship among classes. Hence, for instance, the Organization dimension has a Person level that is part-of a Team or of a Department, which in turn are part-of an Enterprise, and so forth. This ontological description enables usual OLAP operations over indicators, namely roll-up and drill-down. The roll-up operation has a direct correspondence with the part-of relationship, while the drill-down operation implies an opposite path to that described by the part-of.

Finally, the semantics of an indicator cannot be fully given without the representation of its *Formula*, which is the way the indicator is computed [3]. Each formula is characterized by: the related indicator, the aggregation function, the way the formula is presented, the semantics (i.e., the mathematical meaning) of the formula, and references to its components, which are in turn (formulas of) indicators. The formal representation and manipulation of the structure of a formula is essential in Virtual Enterprises to check inconsistencies among independent indicator definitions, reconcile indicators values coming from different sources, and provide the necessary flexibility to indicators management. To this end we need a new kind of ontology, whose peculiarity is the contemporaneous use of logic axioms as well as algebraic formulas to represent the information about indicators. We resort to mathematical standards for describing the way the formula is presented and its semantic (MathML [14]), and for representing operators semantics (OpenMath [15]). Interested readers can refer to [16] for more details about the use of these standards.

4 Reasoning Functionalities

Reasoning about KPIs is mainly based on the capability to manipulate formulas according to strict mathematical axioms, like commutativity, associativity and distributivity of binary operators, and properties of equality needed to solve equations. Therefore, in order to define KPI reasoning functionalities we need to represent both KPIOnto formulas and mathematical axioms in a common logic-based layer. To this end, we refer to the first-order logic and define the functionalities in logic programming (LP); in particular, we use Prolog and we refer to XSB, a logic programming and very efficient deductive database system[2], as reasoning engine. KPIOnto formulas are translated from MathML-Content to LP facts, while we have adapted and extended the Equation Solving System (PRESS) [17] for the representation of mathematical axioms in LP predicates. Indicators' formulas are translated by using two predicates, as follows:

- `formula(Y,Expression,K)` is a LP fact representing the formula related to an instance of the Indicator class in the KPIOnto. Y is the indicator name, Y=Expression is the formula of the indicator written using the infix notation, and K is a constant identifying whether Y is a leaf indicator or not. In the KPIOnto a leaf indicator is an indicator whose formula is not defined on the basis of other indicators.
- `equation(Equation,X)` represents a generic equation in the variable X. When a new indicator has to be added to the knowledge base, at first it is inserted using the equation predicate, and in this form it goes through correctness and consistency checks, as explained below. If the check is positive a new `formula` fact is added to the LP theory, and hence to the ontology.

In the following subsections we introduce the main functionalities for reasoning about KPIs, whereas in Section 5 some usage examples are provided.

[2] `http://xsb.sourceforge.net/`

4.1 Formula Manipulation

Formula Manipulation is an essential reasoning functionality already available in PRESS, that consists in walking through the graph of formulas to derive relations among indicators and rewriting a formula. All the other reasoning functionalities are based on Formula Manipulation. We consider two main types of predicates: for the simplification and for the resolution of equations. The former are used both to compact and to improve the rendering of a formula, by individuating and collecting equivalent terms (e.g., $a*b$ and $b*a$, or $a*a$ and a^2, or $a*1/b$ and a/b), and by deleting unnecessary brackets (e.g., [()]=()). The main predicates of this kind are `simplify_term`, which is used to simplify equations, and `simplify_solution` that rewrites the final solution in a more understandable form.

The second type of predicates enables the symbolic resolution of equations by applying mathematical properties (e.g., commutativity, factorization), and properties of equality. The number and kind of manipulations the reasoner is able to perform depend on the mathematical axioms we describe by means of logical predicates. The resolution of the equation *Equation* in a given variable X starts from the predicate `solve_equation`, that is defined as follows:

```
solve_equation(Equation,X,X=Solution) :-
    solve_equation_1(Equation,X,X=SolutionNotSimplified),
    simplify_solution(SolutionNotSimplified,Solution).
```

where *Solution* is the solution of the equation.

The predicate solve_equation_1 handles different kinds of equations, e.g. linear and polynomial ones[3]. For lack of space, here we report only the method for solving linear equations, as follows:

```
solve_equation_1(LeftMember=RightMember,X,Solution) :-
    simplify_term(LeftMember-RightMember,LeftMember1),
    single_occurrence(X,LeftMember1=0),!,
    position(X,LeftMember1=0,[Side|Position]),
    isolate(Position,LeftMember1=0,X=Solution).
```

The resolution of linear equations is based on the isolation method, consisting in manipulating the equation and trying to isolate the variable X on the left side of the equation; the right side is the requested solution. To this end the equation is firstly simplified by means of the `simplify_term` predicate, then the `single_occurrence` predicate is verified to check whether the equation is linear or not. This predicate `single_occurrence(X,LeftMember1=0)` is true if the variable X is only in one term of *LeftMember1*. Note that after the simplification, terms of the same degree are grouped together, so if the equation is linear the term of first degree of X is present only once. The other predicates are verified to actually implement the isolation method. The `position` predicate returns the side and list of positions of the variable X in the equation. For instance, the side

[3] At the present stage of the project, the formulas of all KPIs described in the KPIOnto are linear and polynomial equations.

and position of x in the equation 3*(1/x)-5*y=0 are respectively "1" (left side) and "1,2,2", i.e. the first position with respect to the minus operator, the second argument of the product, and finally the second argument of the division. At this point, predicates for isolation are used:

```
isolate([N|Position],Equation,IsolatedEquation) :-
    isolax(N,Equation,Equation1),
    isolate(Position,Equation1,IsolatedEquation).
isolate([],Equation,Equation).
```

The set of predicates `isolax` are needed to move a term from a side to another, for instance multiplying or adding the same term to both sides. In total, in order to perform the formula manipulation functionalities more than 900 predicates are used in the theory.

4.2 Equivalence Checking

During KPI elicitation and ontology population, it is useful to individuate and to manage duplications. In our scenario, two KPIs are duplicated if their formulas are equivalent. If duplicates are found, one can choose to erase them leaving only one definition for each KPI in the ontology or to allow multiple definitions by introducing a sort of *same_as* property among KPIs formulas. The second choice is useful to explicitly represent alternatives, to link a formula to the enterprise that actually uses it, and to reduce the time to make inference.

Given two formulas F and G, they are equivalent if F can be rewritten as G, and vice versa, by exploiting Formula Manipulation functionality. The equivalence check is implemented by means of the following predicate:

```
equivalence(Equation,X,Y) :-
    expand_equation(Equation,ExpandedEquation),
    solve_equation(ExpandedEquation,X,X=Solution),
    formula(Y,S,_), expand_equation(Y=S,Y=ES),
    solve_equation(Y=ES,Y,Y=Solution2), Solution=Solution2.
```

where Y is the formula in the ontology that is equivalent to the *Equation* in the variable X.

In order to avoid ambiguity, the check of equivalence is made at leaves level, where indicators are defined without referring to other indicators. Hence, an equation at this level can not be expanded further, and there is no other way to write it. To this end, the `expand_equation` predicate recursively replaces each term in *Equation* with its formula, until we obtain an *ExpandedEquation* that is formed only by leaves indicators. After this transformation the variable X could also be at the right side, then the *solve_equation* is called to rewrite the equation. The same is done for each formula in KPIOnto. If after these rewritings a formula Y exists such that it is equal to the equation, then the `equivalence` predicate is satisfied and the equivalent formula is returned.

4.3 Consistency Checking

When new indicators are added to the ontology or existing indicators are updated or deleted, it is needed to check that such changes do not contradict the ontology. To this end *Consistency Checking* is introduced, that is based on the idea that equivalence relations must be preserved both between indicators and formulas. As a consequence, a formula F_1 for a new indicator I_1 is consistent with the ontology if, whenever there exists a formula F for an indicator I already defined in the KPIOnto such that F_1 can be rewritten as F, then the equivalence $I \equiv I_1$ does not contradict the ontology. Otherwise we have an inconsistency.

In the Prolog theory, we introduced the `incoherence` predicate as follows:

```
incoherence(Equation,X,N) :-
    expand_equation(Equation,ExpandedEquation),
    solve_equation(ExpandedEquation,X,X=Solution),
    formula(Y,S,_,N), expand_equation(Y=S,Y=ES),
    solve_equation(Y=ES,X,X=Solution2), Solution S̄olution2.
incoherence(Equation,X,incosistent) :-
    expand_equation(Equation,ExpandedEquation),
    solve_equation(ExpandedEquation,X,X=0).
incoherence(Equation,X,N) :-
    \+expand_equation(Equation,ExpandedEquation),
    formula(X,_,branch_node,N),!.
```

Given an equation (X=*Equation*), the predicate returns whether the equation is consistent with the ontology and, if any, the reason of the inconsistency. The `incoherence` predicate is based on the definition of equivalence, as described above. Given the set of formulas in the ontology that can be expanded and rewritten as functions of X, each element that is different from the given *Equation* (after the appropriate expansion) leads to inconsistency and is returned. We have an inconsistency also when, after a manipulation, the new equation assumes the form X=*0*, or when the new equation cannot be expanded but a non-leaf formula for X exists. In other cases the consistency is verified.

4.4 Extraction of Common Indicators

The last reasoning functionality we present in this work is the extraction of common indicators. Given a set of indicators $\phi = \{I_1, I_2, ..., I_n\}$, common indicators of ϕ (denoted by ci(ϕ)) is the minimal set of atomic indicators needed to compute all formulas of ϕ. A more general notion refers to the extraction of the k-Nearest Neighbours of a set of indicators ϕ (denoted by NN(ϕ,k)), that is, the set of indicators at a distance k from all indicators in ϕ, where the distance is the minimal number of dependency arcs separating two indicators. Note that ci(ϕ)=NN(ϕ,$+\infty$). Common indicators and k-Nearest Neighbours are useful to determine the possible sets of information that have to be provided by enterprises in order to calculate the requested KPI or, in a bottom-up fashion, to recognize what kind of indicators can be derived, given the available information at enterprise level. Hence, they are essential tools at design and measuring time. On the basis of these definitions, we have implemented the `meaToInd` predicate which, given a set of indicators (called measures), returns the derivable KPIs and `indToMea` predicate that, given a set of KPIs, returns the sets of measures needed to compute all the KPIs. These two predicates extend the definition of k-Nearest Neighbours, by returning the set ϕ for any values of the distance k.

```
indToMea(Ent,L,Lfin):-
     hasEntInd(Ent,[L],[],Lfin2), sort(Lfin2,Lfin).
```

where *Ent* is the name of the enterprise, *L* is the set of KPIs and *Lfin* the set of needed measures.

The predicate is satisfied for each set *Lfin* of indicators provided by the enterprise, such that a formula for *L* can be derived from the ontology and it is defined on the basis of this set. `hasEntInd` checks, by using Formula Manipulation predicates, that an indicator is provided by the enterprise.

```
meaToInd(Ent,Lin,V):-
     hasMeaEnt(Ent,Lin), formula(V,F,Kind,Fname),
     \+member(V,Lin), indToMea(Ent,V,L), subset(L,Lin).
```

where *Lin* is the set of measures and *V* the set of indicators derivable by *Lin*.

As first step, the predicate `hasMeaEnt` is called to check whether *Ent* provides all indicators in *Lin*. Now, by using `indtoMea`, the predicate returns each indicator *V* that can be computed by a subset of *Lin*. If *Ent* is not specified, then the predicate is checked for all the enterprises.

5 Case Study

Let us consider two partners, ACME and ACME2, willing to build up a Virtual Enterprise. They agree on the KPIs to be used to monitor the VE, defined by the following formulas, whose structural relations are depicted in Figure 2:
- $PersonnelCosts = NumHours * HourlyCost * (Overhead + 1)$
- $Costs = TravelCosts + PersonnelCosts$
- $PersonnelTrainingCosts = HourlyCost * NumTrainingHours$
- $TeachCost = NumTrainingHours * HourRate$
- $InvestmentInEmployeeDevelopment = PersonnelTrainingCosts + TeachCost$

At one point, ACME realizes that would need other KPIs for monitoring the VE and proposes to introduce two new indicators (1) *IG_ROI* and (2) *TotCostsEmpTrain*, currently not defined, which are aimed to measure respectively the Return On Investment related to those ideas generated by the organization and the total internal costs invested for training of employers:

$$ROI\_IG = ExpectedMarketImpact/Costs$$

$$TotCostsEmpTrain = TeachCost + NumTrainingHours * HourlyCost$$

After having translated such KPI formulas in Prolog facts, the Equivalence and Consistency Checking functionalities are used to verify whether formulas

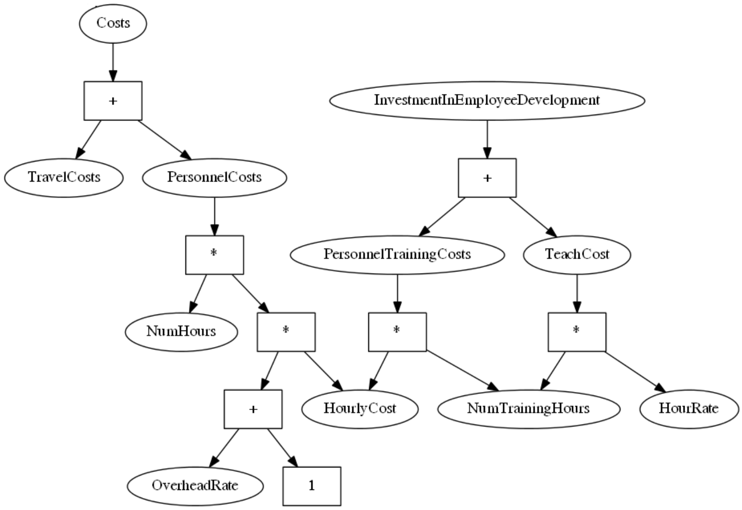


Fig. 2. Formula tree for the example

are duplications of existing ones and whether they do not make the ontology inconsistent. Given that the *ROI_IG* formula is new and does not contradict any previously defined formula, it is added to the Prolog theory. On the contrary, by using the formula about *PersonnelTrainingCosts* and after some formula manipulation, the reasoner discovers that *TotCostsEmpTrain* is structurally equivalent to *InvestmentInEmployeeDevelopment*. Hence, ACME can refer to this last one to annotate its data without introducing a new indicator.

Consistency Checking is exploited also when an enterprise tries to update the definition of a KPI. For instance, let us consider that ACME requires to update *NumTrainingHours*, committing a manifest error when typing the formula:

$NumTrainingHours = PersonnelTrainingCosts + HourlyCost$

In this case, ACME discovers that the change directly contradicts the definition of *PersonnelTrainingCost* together with every other higher-level indicators depending on *PersonnelTrainingCost*, i.e. in this case *InvestmentInEmployeeDevelopment*.

By using the `meaToInd` predicate, we can introduce another way to support the choice of the KPIs to use at VE level. In particular, starting from those indicators that each enterprise chooses to share in the VE, we are able to discover the common set of KPIs the VE is able to globally compute. In the present case study, we assume that ACME and ACME2 respectively share in the VE data about the following set of indicators:

ACME={*ROI_IG*, *InvestmentInEmployeeDevelopment*, *HourlyCost*, *NumHours*, *OverheadRate*}

ACME2={*HourRate*, *NumHours*, *NumTrainingHours*, *OverheadRate*, *PersonnelTrainingCosts*, *ExpectedMarketImpact*, *TravelCosts*}

The Extraction of Common Indicator functionality returns the set of all the computable KPIs for the VE: {*OverheadRate*, *NumHours*, *HourlyCost*, *Person-*

nelCosts, *ROI_IG*}. We'd like to note that the last three of such indicators are directly computable only by ACME. However, ACME2 provides *NumTrainingHours* and *PersonnelTrainingCosts*, from which the reasoner is able to derive *HourlyCosts* (by inverting the formula of *PersonnelTrainingCosts*), and then *PersonnelCosts* and also *ROI_IG*.

In order to actually compute KPIs values during the monitoring of the VE, and also in case of the introduction of a new indicator, not previously defined, the functionality provided by the *indToMea* predicate can be exploited. It is used to verify whether it is possible to evaluate the required new indicator starting from the indicators provided by the enterprises. As a by-product, it retrieves the list of indicators that, for each enterprise, are needed for actually computing the requested KPIs. For instance, if the value of *ROI_IG* is required, the reasoner reports as output the following set of indicators:

ACME={*ROI_IG*}

ACME2={*NumHours*, *NumTrainingHours*, *OverheadRate*, *PersonnelTrainingCosts*, *ExpectedMarketImpact*, *TravelCosts*}

If a KPI can not be computed from the provided indicators, then the reasoner is able to return the minimal list of additional indicators to be provided.

6 Conclusions

In this work we introduced a semantic model for the definition of KPIs from both a conceptual and a structural perspective. The KPIOnto ontology is exploited in the BIVEE project as common conceptual reference for annotation of KPIs about real enterprise data, whereas a logic-based formalization in Prolog of the structure of KPI formulas is useful to support various reasoning functionalities. Although those introduced in this work are especially suited for the setup of a VE, others have been described in a previous work for execution of OLAP-like and complex queries [16]. The proposed functionalities are currently exposed through web service interfaces, together with other reasoning services of the PIKR, i.e. the semantic layer of the BIVEE project. In this way those BIVEE modules appointed to offer high-level functions and graphical user interfaces to business managers can use the services in a loosely-coupled fashion. Here we focused only on linear equations and common aggregation functions (i.e., minimum, maximum, average, sum or count), because the more than 200 KPIs provided by the enterprises within the BIVEE project do not require more complex representations. However, more predicates are already available to deal with generic non-linear equations, that can be added to the theory thus increasing both expressiveness and computational complexity. In order to keep under control this last, which is often critic in logic programming, we are both analytically and empirically evaluating the theory, in order to locate bottlenecks, useful to implement optimizations strategies by refactoring LP predicates. As future work we also plan to conduct tests on data coming from real use cases, to fully integrate both already existing and new functionalities with all the other modules of the project, and to extensively populate the ontology.

References

1. Diamantini, C., Potena, D., Proietti, M., Smith, F., Storti, E., Taglino, F.: A semantic framework for knowledge management in virtual innovation factories. Int. Journal of Information System Modeling and Design (IJISMD) (in press)
2. Tseng, F.S., Chen, C.W.: Integrating heterogeneous data warehouses using xml technologies. J. Inf. Sci. 31(3), 209–229 (2005)
3. Diamantini, C., Potena, D.: Thinking structurally helps business intelligence design. In: D'Atri, A., Ferrara, M., George, J.F., Spagnoletti, P. (eds.) Information Technology and Innovation Trends in Organizations, pp. 109–116. Physica-Verlag HD (2011)
4. OECD: Glossary for oecd composite leading indicators, `http://www.oecd.org/document/1/0,3746,en_2649_34349_32694145_1_1_1_1,00.html`
5. Adams, R., Bessant, J., Phelps, R.: Innovation management measurement: A review. International Journal of Management Reviews 8(1), 21–47 (2006)
6. Zheng, H., Chanaron, J.J., You, J., Chen, X.: Designing a key performance indicator system for technological innovation audit at firm's level: A framework and an empirical study. In: IEEE International Conference on Industrial Engineering and Engineering Management, IEEM 2009, pp. 1–5 (December 2009)
7. del-Río-Ortega, A., Resinas, M., Ruiz-Cortés, A.: Defining process performance indicators: An ontological approach. In: Meersman, R., Dillon, T.S., Herrero, P. (eds.) OTM 2010. LNCS, vol. 6426, pp. 555–572. Springer, Heidelberg (2010)
8. Supply-Chain Council: Supply chain operation reference (scor) model, overview-version 10.0., `http://supply-chain.org/`
9. Value-Chain Group: Value reference model (vrm), `http://www.value-chain.org/en/cms/1960`
10. Tennant, G.: Six Sigma: SPC and TQM in manufacturing and services. Gower Publishing, Ltd. (2001)
11. Measurement Units Ontology: `http://forge.morfeoproject.org/wiki_en/index.php/units_of_measurement_ontology`
12. Kimball, R., Ross, M.: The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, 2nd edn. John Wiley & Sons, Inc., New York (2002)
13. Neumayr, B., Schrefl, M.: Multi-level conceptual modeling and OWL. In: Heuser, C.A., Pernul, G. (eds.) ER 2009 Workshops. LNCS, vol. 5833, pp. 189–199. Springer, Heidelberg (2009)
14. W3C Math Working Group: Mathematical Markup Language (MathML) Version 2.0., 2nd edn., `http://www.w3.org/Math/`
15. OpenMath Society: The OpenMath Standard Ver. 2.0., `http://www.openmath.org/`
16. Diamantini, C., Potena, D.: Semantic enrichment of strategic datacubes. In: Proc. of the ACM 11th Int. Workshop on Data Warehousing and OLAP, pp. 81–88. ACM (2008)
17. Sterling, L., Bundy, A., Byrd, L., O'Keefe, R., Silver, B.: Solving symbolic equations with press. J. Symb. Comput. 7(1), 71–84 (1989)

Cross-Domain Crawling for Innovation

Pierluigi Assogna and Francesco Taglino

Istituto di Analisi dei Sistemi ed Informatica “A. Ruberti” - IASI-CNR
Viale Manzoni 30, 00185 Roma, Italy
pierluigi.assogna@mensa.it, taglino@iasi.cnr.it

Abstract. Innovations, in any field, originate in the mind of people, on the base of mechanisms not yet completely understood. There have been many studies relevant to thinking techniques that have been proven to favor creativity, like for instance those studied by De Bono. A general characteristic of these techniques is the recommendation of avoiding usual thinking paths, habitual mind frames: this is facilitated by putting oneself in unusual physical settings, or introducing absurd concepts, and the like. The use of metaphors is another recognized enabler of creativity, by bridging different conceptual domains.

A Knowledge Base (KB) structured around an Ontology can be seen as a close simulation of the conceptual structure that, according to Constructivism, supports a person’s thinking processes, and the Web can be seen as the corresponding world to be explored and that contributes to that person’s culture. This kind of domain specific KBs is being organized and used as support for advanced enterprise information systems. This paper presents a technique for extending the working domain (WD) of an organization with concepts belonging to other domains, obtained by retrieving documents that discuss both concepts of this WD and “foreign “ ones. These documents, proposed to the KB editors, are considered candidates for innovative problem solving activities and considerations.

Keywords: Ontology, Web Crawling, Innovation, Creativity.

1 Introduction

Innovations, in any organization, can of course only come from the people involved there, and not from the systems. Systems can support all the stages of the process of innovation, as for instance analyzed in the BIVEE project[1]. The first spark that triggers an innovation has not yet found effective support: there are no established ICT methods and tools specifically focused on this starting discontinuity.

As an example of the SotA in the field of creativity support, we can see that within the European Objective 8.1, 'Technologies and Scientific Foundations in the field of creativity' [1] we can find, at their start, the development of 3 platforms (i-Treasures, CULTAR, RePlay) for supporting the use of cultural resources, 2 projects targeted at promoting creativity: Idea Garden [2] (planning to develop hard and software

[1] http://wordpress.bivee.eu

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 286–297, 2013.

technologies that assist designers during all phases of the creative process), and Collage [3] (planning to develop an innovative Social Creativity Service-Set), plus a number of systems aimed at capturing, maintaining, providing a structured view of cultural heritage.

The implicit assumption for these projects is: the main ingredient for supporting the process of creation is the availability of existing knowledge, that needs to be adequately proposed to (human) designers.

The method proposed in these notes aims at providing a pre-digested form of this ingredient, simulating the processes allegedly going on (according to recent studies) in our minds when we are confronted by new issues. Along this line the European Commission launched in 2009 the Human Brain Project, aimed at analyzing the thinking processes of the mind. One of the objectives of it is to "build revolutionary new computing technologies" [4].

A Knowledge Base (KB), together with the supporting semantic services, should simulate as close as possible the architecture and workings of our mind, in order to be an effective support to our thinking and decision making processes, like a sort of super-fast Thinking Assistant. There are two big issues related to this approach: a) we do not really know how a mind works at this abstract level, and b) what we know for sure is that our mind is probably the most complex system that we know of in the universe. Luckily both issues, rather than discourage research and commercial efforts, keep driving a substantial body of researchers and stakeholders, and promoting an interesting convergence of neural scientists and ICT experts.

This paper presents a research hypothesis that combines accredited theories of mental processes with state of the art knowledge -management and -mining technologies, as a support for creative, innovative thinking.

2 Cognitive Issues

2.1 Construction of Conceptual Structures

From Wikipedia we get a good definition for Constructivism: "… is a theory of learning and an approach to education that lays emphasis on the ways that people create meaning of the world through a series of individual constructs". According to this theory, that is one of the most accredited, our mind, our personal culture, is a dynamic (generally growing) network of concepts and models, that starting from a limited set genetically inherited, keeps getting more and more complex during our life. The "father" of Constructivism is generally considered Jean Piaget [5].

The models are used to organize experiences, that in turn are used to create more models. This paradigm is very distant from Locke's *tabula rasa*, as it presumes the existence of a first set of models, and is in tune with Kant's *a priori* categories. An exaggeration of this conception is the stereotype that you find easily just what you are looking for.

An interesting theory, Memetics, now dwindling out of popularity, is a reductionist approach related to knowledge acquisition, proposed by Richard Dawkins in his book "the Selfish Gene" [6], where concepts and methods that are experienced, exchanged, learnt by people migrate from mind to mind like genes across DNA's strands, so that Darwinian mechanisms can be applied to them.

It is of particular interest that two theories stemming from two fiercely opposed approaches (teleology vs causality) can both provide hints for the proposed simulation of conceptual processes related to creative thinking. Constructivism because of its focus on the importance of models for knowledge acquisition and integration, Memetics because of its "objectification" of concepts and models, and of the underlining idea that "memes" are capable of auto-organization.

The stage of the proposed simulation is composed of:

- Knowledge Bases and services seen as a computerized version of minds, in terms of evolution and usage
- The conceptual structure, represented by one or more ontologies, populated with topics (and related content) linked to each other
- The net as the world to be experienced (searched)
- The auto-organizing capability of mind, represented by the application of specific methods for the semantic annotations (connections) that characterize each conceptual node
- Conscience, that In orthodox Memetics has no (or very scarce) place but in our case has a strong role, represented by the stakeholders (contributors/users) of the KB. The semantic annotations that each contributor is asked to apply to an inserted topic (concept, class, method, process, etc.) make up the overall structure, the KB's culture.

The main challenge is the possibility of simulating the (supposed) auto-organizing capability of memes in minds. In this way a KB could exploit the enormous corpus of knowledge items present in the net, taking advantage of the searching speed of bots.

The focus of these notes is innovation, so that as first step we need to consider the mind's functionalities that seem to promote innovation, and then proposing how to simulate them.

2.2 Lateral Thinking and Metaphors

Lateral Thinking is an expression coined in 1967 by Edward de Bono [7], an author that has devoted much of his work to problem solving methodologies. Even if criticized, his ideas have had considerable impact on the methods aimed at finding creative solutions to all sort of problems. The basic approach is that of figuratively "stepping out" of a top-down or bottom-up rational thinking process, when faced by an issue, and looking at the situation from a lateral view point, that is from a "path" of the conceptual structure that is not strictly connected to the ones facing the issue. An example of this approach is opening a dictionary at a random page, taking a random item, and seeing how it applies to the issue at hand.

If we consider the "world of ideas" represented by the sum of all human cultures as an interconnected conceptual structure, this technique means associating, on an ad-hoc mode, topics pertaining to remote sections of the structure, while each singular mind hosts connected memes that belong to specific limited domains.

It is widely accepted that innovations, from art to technology to any other field, come by trodding scarcely or never used paths connecting concepts, methods, processes, situated in distant branches of knowledge structures.

An interesting experiment of a semi-automated proposition of heterodox connections is represented by the “metaphorical search engine” experimentally provided by the organization YossarianLives![2]: the idea of their approach is that while traditional search engines tend to retrieve documents related more or less closely to the keywords entered, consolidating in this way the implied conceptual domain, their metaphorical search retrieves documents that are metaphorically connected, that hence could help the searcher to address “laterally” the issues he/she is affording.

Metaphors are a very powerful way for triggering innovative and out-of-the-box ideas in relation to an issue. Their use can be seen as a form of lateral thinking that starts from a neighborhood that, even being different, has some subtle relations with the one where the issue starts.

The first authors to point out the fact that our common language is ripe of metaphors have been Lakoff and Johnson [8]. They analysed the characteristics of a metaphor. It connects two conceptual domains, termed “source” and “target”, and exercises a conceptual transfer between the two. In order to be effective, that is to trigger new perspectives and conceptual paths, it has to comply to the so-called “Invariance Hypothesis”: the mapping can exercise its suggestion if it is applied to similar “image-schemas” [9] on both sides of the connection. These image-schemas (examples proposed by Johnson are: Merging, Matching, Iterating, Splitting, and the like) can be interpreted as processes.

Many phenomena share similar evolution processes, in particular those related to complex environments. Their direct comparison can provide useful hints: if you draw parallels between drivers, trends, obstacles, enablers, anything that characterizes the stages of these processes, you may find that specific situations, that are clearly outstanding in process X, can help in shading light to a parallel situation in process Y, where these are not easy to pinpoint, and this can boost innovative considerations and actions.

2.3 Automatic Brain Activities

Neurological researchers, particularly through the recent use of brain imaging techniques, have demonstrated that in experiments (see for instance [10]) where a person is asked to perform simple actions requiring a rational choice (like pushing a button every time a specific sound scheme is perceived) the conscious decision related to the action is always registered in the brain seconds after the action has been automatically performed. This outcome has been erroneously used by many as a proof that free will (or conscience) does not exist. A person who plays a musical instrument, let’s take a piano, has always been aware of this situation: you have to pay attention to the keys you press only in the early study stage of a specific piece; when you have learnt it, if while you are playing you happen to think of the keys you are pressing, a mistake is

[2] www.yossarianlives.com

guaranteed. This simply means that a task that initially requires a conscious decision process, is by and by delegated to lower level processes. In this case the functions that typically maintain the request for attention are those like the loudness, and the expression you want to communicate to the audience.

In any case we know that most of the activities of our brain are automatic. In the case of a creative thinking process you have a conscious tension, that leads you to walk many paths, some of which have been built automatically. There are many stories of solutions found while sleeping, so also the exploration can go on automatically.

Artists, for instance, keep integrating into their conceptual structure bits and pieces of esthetic memes, and this happens because they consciously at first, and automatically with practice, know that any of these could become useful sometime. These are then used, when creating an artifact, as if they were there by chance or by an overall automatism.

2.4 Extensions of Domain Knowledge Bases

A person can master a very limited part of the global knowledge, and along the same line an organization maintains, in order to provide a semantic structure for its activities, a domain-oriented Knowledge Base, as it would be unreasonable and unuseful to maintain a "universal" base.

In this case *domain* means not only a bounded set of concepts, but also a local, subjective interpretation and appreciation of each concept. In fact the KB is mainly used by humans, and in this respect the ambiguity of natural language, that is the main communication mode among minds, rather than be considered as a disadvantage, is to be seen as the main source of creativity and explosion of the global culture.

In this respect the considerations related to the role of *boundary objects* within a specific organization (see for instance [11]) underline the fact that different functional units can assign different semantic value to the same concept, and the "boundary" documents that are exchanged between them have the potential at the same time of causing misunderstandings or of promoting innovative ideas. The considerations presented in this Paper explore the value of crossing more drastic boundaries, those existing between disconnected domains, where the melting and contrasting of heterogeneous concepts can trigger innovation.

Most, if not all, of the topics that populate the typical domain KBs are relevant to the organization's mission. Additional and "diverging" knowledge is typically imported by its users, either by following specific searches related to any new situation or important event, or through the use of tools like RSS subscriptions or "follow X". In any case these "institutional" enrichments do not generally tap into different domains of interest. This mode of using the existing knowledge is analogous to a person using his/her settled personal models, skills, experience, to manage events. There are cases when a person has to update his/her models, when for instance an experience is not easily processed using them, and it is anyway considered important. In these cases the flexibility of modifying one's models can be very important for taking benefit from these experiences. The creation of new models requires the integration, into one's conceptual structure, of new memes, that are concepts, processes, and so on, pertaining to different domains. And this has to be done ad-hoc, when required, as it is by definition impossible to forecast the unexpected.

It would be preposterous for a person to think of enriching his/her culture in all directions, and with enough depth, to be prepared for any unexpected situation. What is needed is a) the predisposition to change, b) the availability of tools for mining useful pieces of knowledge in domains that are related to the new situation that needs to be afforded (the "know who knows what" approach), and c) the capability of using them efficiently. The same works for organizations: their KBs have to be focused on their usual business, and the related ontologies management systems must include mechanisms ready for supporting ad-hoc mining activities.

A way that a person has, in order to be prepared for these possible sudden requirements for a fast integration of substantial chunks of new topics and models (like for instance when moving from a city to another one, may be in another Country) is to maintain a basic knowledge related to domains "neighboring" to the one this person is managing, as the probability of having to explore a different conceptual sector is higher for contiguous rather than for totally removed ones. This knowledge not focused on one's daily interests can be called lateral, also in view of its potential capability of triggering lateral thinking in day-by-day tasks. The problem here is that even limiting the number of candidate contiguous domains, this number can be very high, so that subjective selection methods are generally applied.

An organization, on its part, needs only tools for exploring the net when the need to expand its knowledge arises, that is when a creative solution for an issue is required.

3 Ontology-Guided Crawlers

On the base of the considerations that ontologies can represent the "conscience" of an organization, and that the net represents the world to explore, we propose to automate some of the steps of the process that we have described above, in order to support the search of documental resources in extended and loosely related domains.

An option that we are exploring is based on the organization of a crawler capable of searching documents that in specific ways (controlled by the organization's stakeholders) are related to an issue, or are considered candidate for triggering new ideas relevant to the organization business.

In particular, here we outline a semantics-based search method that integrates automatic and human controlled steps (Fig. 1). For knowledge extraction from text, we are currently using Alchemy[3], a set of APIs able to analyze textual documents by using sophisticated statistical algorithms and natural language processing technologies. A sketchy architecture of the infrastructure that implements the method is reported in Fig 1 and described in terms of its main components as follows.

[3] `http://www.alchemyapi.com/`

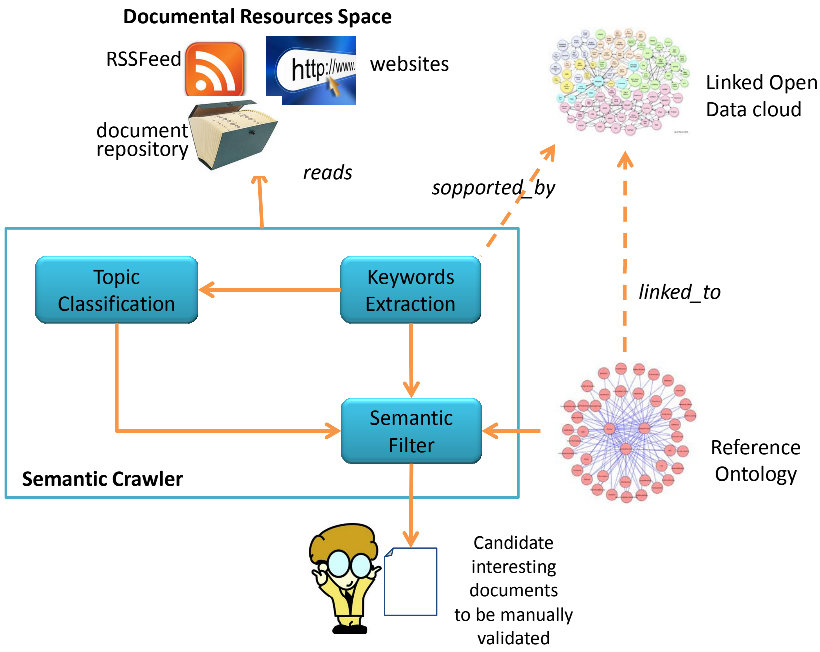


Fig. 1. Semantics-based cross-domains crawling

Documental Resources Space: This is the space where we search for interesting documents. Here, we can have different kinds of resources, for instance websites, rss feed, and public documents repositories. As illustrated later in the examples, our reference domain is Robotics and Machine Vision (R&MV). So, we could search in very technology- and innovation-oriented sources (e.g., the section of the MIT website on innovations[4]), as well as in generalist websites like the BBC news portal[5].

Bridging the Reference Ontology and the Linked Open Data Cloud. As a preliminary step we assume to have a reference ontology (or at least a vocabulary) specific for our domain of interest, e.g., Robotics and Machine Vision (R&VM) and to associate it to contents in the Linked Open Data[6] cloud. For instance, in the BIVEE project, we currently have built a glossary of almost 600 entries about Robotics and Machine Vision. In this first implementation, for each entry in our glossary we tried to identify a corresponding entry in the DBpedia dataset (DBpedia is a project aiming to extract structured content from the information created as part of Wikipedia). If the entry exists we created a link between the entries via the *owl:sameAs* property. For instance, the *Photodiodes* and *Camera* entries in our R&MV glossary have been

[4] http://web.mit.edu/newsoffice/topic/innovation.html
[5] http://www.bbc.co.uk/news/
[6] http://www.w3.org/DesignIssues/LinkedData

linked to the *Photodiode* (http://dbpedia.org/page/Photodiode) and *Camera* (http://dbpedia.org/page/Camera) entries in DBpedia, respectively.

Terms Extraction: This module is in charge of extracting relevant terms from a given document. Relevant terms are those which are intended to be representative and somehow synthesize the document's content. For each extracted keyword, the *URLGetRankedConcepts* method from the AlchemyAPI, which has been used for the keywords extraction, reports the reference to the corresponding DBpedia entry, and a numeric value in the range [0..1] representing the relevance of the keyword with respect to the document.

Topic Classification: This module is in charge of classifying candidate interesting documents with respect to their main topic. The *URLGetCategory* from the AlchemyAPI is able to classify documents in one of the following categories: *Arts & Entertainment*, *Business*, *Computers & Internet*, *Culture & Politics*, *Gaming*, *Health*, *Law & Crime*, *Religion*, *Recreation*, *Science & Technology*, *Sports*, and *Weather*. Since they are high level categories, extracted terms could help in refining the classification.

Semantic Filter

Once keywords have been extracted, the semantic filter is in charge of proposing if the analyzed document is an interesting resource. In this proposal, we define a metrics based on the relevance value of each extracted keyword. First we need to identify extracted keywords that are related to our domain of interest (e.g., Robotics and Machine Vision). For collecting keywords related to the specific domain of interest, for each keyword, one of the two criteria must be satisfied:

- it exists an entry in the domain specific glossary that is *owl:sameAs* the DBpedia entry representative of the extracted keyword. For instance, considering the example related to Document 1 in Table 1, the extracted keyword *Camera* is considered in the domain of R&MV because the *Camera* entry is in the R&MV glossary and it has been previously linked to the *Camera* entry in DBpedia.
- it exists an entry in the domain specific glossary that has been defined *owl:sameAs* a given DBpedia entry, and this DBpedia entry has in common with the DBpedia entry representative of the extracted keyword a subject. For most of the DBpedia entries a set of subjects is defined through the Dublin Core *subject* property. For instance, both *Photodiode* and *Light-emitting-diode* entries in DBpedia have *optical_diodes* among their subjects. For this reason, considering the example related to Document 1 in Table 1, the extracted keyword *Light-emitting-diode* is considered to be related to the R&MV.

Table 1. Example of anlyzed articles from the BBC news web portal

Document 1	http://news.bbc.co.uk/2/hi/science/nature/1542588.stm			
Classification	science_technology		**R&MV**	**Other**
Text and **Relevance**	Light-emitting diode(*)	0.913929	**0.37**	**0.48**
	Slug	0.858322		
	Power (*)	0.855041		
	Foot-and-mouth disease	0.854825		
	Mucus	0.832959		
	Camera(*)	0.831103		
	Soil	0.830792		
Document 2	http://www.bbc.co.uk/news/technology-10687701			
Classification	computer_internet		**R&MV**	**Other**
Text and **Relevance**	Female body shape	0.967518	**0.11**	**0.40**
	Body shape	0.635835		
	Clothing	0.476781		
	Human body	0.467204		
	Robotics(*)	0.447413		
	Robot(*)	0.441914		
	Fashion	0.342003		
	Physical attractiveness	0.331898		
Document 3	http://www.bbc.co.uk/news/health-21965092			
Classification	Health		**R&MV**	**Other**
Text and **Relevance**	Obesity	0.976225	**0**	**0.54**
	Bariatric surgery	0.597715		
	Gastric bypass surgery	0.535516		
	Weight loss	0.479716		
	Bacteria	0.464552		
	Nutrition	0.45946		
	Bariatrics	0.45838		
	Medicine	0.374071		
Document 4	http://www.bbc.co.uk/news/business-20800118			
Cassification	arts_entertainment		**R&MVs**	**Other**
Text and **Relevanc**	Robot(*)	0.971036	**0.42**	**0.14**
	Robotics(*)	0.691485		
	White-collar worker	0.615792		
	Industrial robot(*)	0.509681		
	Humanoid robot(*)	0.418013		
	Manufacturing	0.37688		
	Automaton(*)	0.347331		

Once we have identified those extracted keywords realted to the specific domain of interest, a document is considered a valid candidate if the sum of the relevance values for the extracted keywords belonging to our specific domain divided by the number of extracted keywords is greater than 0.1 and less than 0.4. At the same time, the sum of the relevance values for the extracted keywords not belonging to our specific domain divided by the number of extracted keywords must greater or equal to than 0.4. The first condition ensures that the analyzed document deals with our reference domain, but in a small manner, while the second constraint ensures that the analyzed document deals with other topics in a considerable measure. According to that, we report in Table 1 the results related to four analyzed articles from the BBC news web portal. Keywords extracted from the documents are marked with an asterisk (*) if they are considered to belong to the R&MV domain.

- Document 1 and Document 2 are considered relevant. This meets exactly our expectations since they consider Robotics and Machine Vision in very singular applications: one for extracting energy from insects and the other for supporting to help shoppers get the right fit when buying clothes online.
- Document 3 is not considered relevant because it does not consider Robotics and Machine Vision at all.
- Document 4 is too much Robotics oriented, so it can be surely useful for experts in the Robotics field, but it does not appear too inspiring for lateral thinking activities.

4 Related Works

Slug[7] is a web crawler designed for harvesting semantic web content. Implemented in Java using the Jena API, Slug provides a configurable, modular framework that allows a great degree of flexibility in configuring the retrieval, processing and storage of harvested content. The framework provides an RDF vocabulary for describing crawler configurations and collects metadata concerning crawling activity. Crawler metadata allows for reporting and analysis of crawling progress, as well as more efficient retrieval through the storage of HTTP caching data.

LDSpider[8] includes handlers to read RDF/XML, N-TRIPLES and N-QUADS. Simple interface design to implement own handlers (e.g. to handle additional formats). Different crawling strategies: Breadth-first crawl; Depth-first crawl; optionally crawl schema information (TBox). Crawling scope: crawl can easily be restricted to specific pages e.g. pages with a specific domain prefix. The crawled data can be written in various ways, such as RDF/XML or N-QUADS. The crawler can write all statements to a Triple Store using SPARQL/Update. Optionally uses named graphs to structure the written statements by their source page. Optionally, the output include provenance information.

APERTURE[9] crawls file systems, websites, mail boxes and mail servers. It extract full-text and metadata from many common file formats.

The major concern about these crawlers is the fact that the searching strategy cannot be guided in terms of a domain specific ontology. However, they will be better investigating for understanding the opportunity to integrating them.

---

[7] `http://ldodds.com/projects/slug/`

[8] `http://code.google.com/p/ldspider/`

[9] `http://aperture.sourceforge.net/index.html`

5 Conclusions and Future Works

The outlined method wants to be a very preliminary work showing that, with existing semantic tools, it is possible to support people in exploiting our mind's capability of cross-fertilizing processes typical of a specific domain with concepts pertaining to a different one, obtaining in this way new insights and viewpoints.

The key concept of the proposed foray for new ideas in the Web is searching for knowledge resources that are not completely centred on one's principal domain of interest, but concerned about both this domain and one or more other ones: the assumption is that a document with these characteristics has a high probability of talking about applications that span these domains, or of using interesting analogies or metaphors connecting them.

The work is in a very early stage and several feature can be improved. About knowledge extraction from documents, we are investigating additional solutions with respect to AlchemiAPI, such as Open Calais[10], Zemanta[11], and Fise[12], considering also the opportunity to integrate more than one single tool. About bridging a domain specific ontology and the contents in the Linked Open Data cloud, we are investigating on using additional ways and not only the *owl:sameAs* property. For instance we could use the generalization/specialization relationships (via the *rdfs:subClassOf* property), and weighting types of links and distances between DBpedia (and not only) entries. Then we also need to massively test the method on a significant number of documents and evaluating at least the precision of the method.

Acknowledgments. This work has been partly funded by the European Commission through the ICT Project BIVEE: Business Innovation and Virtual Enterprise Environment (No. FoF-ICT-2011.7.3-285746). The authors wish to acknowledge the Commission for its support. We also wish to acknowledge our gratitude and appreciation to all BIVEE project partners for their contribution during the development of various ideas and concepts presented in this paper.

References

1. Technologies and Scientific Foundations in the field of creativity (March 01, 2013), http://cordis.europa.eu/fp7/ict/creativity/creativity-objectives_en.html
2. Idea Garden (March 01, 2013), http://idea-garden.org/
3. Collage (March 01, 2013), http://projectcollage.eu/
4. Human Brain Project (March 01, 2013), http://www.humanbrainproject.eu/vision.html
5. Jean Piaget (March 01, 2013), http://en.wikipedia.org/wiki/Jean_Piaget

[10] http://www.opencalais.com/
[11] http://www.zemanta.com/
[12] http://wiki.iks-project.eu/index.php/FISE

6. Dawkins, R.: The Selfish Gene. Oxford University Press. New York City (1976) ISBN 0-19-286092-5
7. de Bono, E.: The Use of Lateral Thinking (1967) ISBN 0-14-013788-2
8. Lakoff, G., Johnson, M.: Metaphors We Live By. University of Chicago Press, Chicago (1980)
9. Johnson, M.: The Body in the Mind: The Bodily Basis of Meaning, Imagination, and Reason, University of Chicago (1987)
10. Unconscious determinants of free decisions in the human brain (March 01, 2013), `http://www.nature.com/neuro/journal/v11/n5/full/nn.2012.html`
11. Carlile, P.R.: A Pragmatic View of Knowledge and Boundaries: Boundary Objects in New Product Development. Organization Science 13(4), 442–455 (2002), doi:10.1287/orsc.13.4.442.2953

Hybrid Modelling with ADOxx: Virtual Enterprise Interoperability Using Meta Models

Robert Woitsch

BOC Asset Management, Operngasse 20b, 1040 Vienna, Austria
robert.woitsch@boc-eu.com

Abstract. This practical paper introduces hybrid modelling and its application in supporting interoperability within virtual enterprises. Based on a survey report of the FINES cluster, different dimensions of enterprise interoperability are introduced before concept modelling as an instrument and meta-modelling as the technological approach is discussed. The challenge of holistically combining different modelling approaches concerned with enterprise interoperability can be tackled via hybrid modelling. The open development platform ADOxx is introduced as a technological basis supporting realization of the hybrid modelling and the semantic lifting. Hybrid modelling applied in the project BIVEE to holistically model a Value Production Space is introduced to demonstrate a complex meta-modelling environment.

Keywords: Meta Models, Concept Modelling, Conceptual Integration.

1 Introduction

Industry 4.0 [1] describes the ongoing paradigm shift in production industry towards networked enterprises. Future Internet Enterprise Systems (FInES), virtual enterprises and networked IT-infrastructure are keywords indicating current and upcoming challenges of future enterprises facing agility, sensing, community-orientation, liquidity and globalism. Enterprise Interoperability (EI) is hence defined "as a field of activity with the aim to improve the manner in which enterprises, by means of ICT, interoperate with other enterprises..." [2]. Concept models such as business processes, value process chains or e3value models are commodity and hence a promising candidate to support EI. A well-known realization approach for modelling is the meta-model technology that is an abstraction of models enabling the domain-specific realization of modelling languages. Depending on the complexity of the requirements, most likely interplay of several meta-models focusing on different viewpoints is required. This interplay between meta-models is seen as hybrid modelling, as different viewpoints are commonly applied to create a holistic observation This short introduction is followed by a section which focuses on different levels of interoperability, starting with a state of the art survey and mapping it to the proposed approach – the concept modelling. Third section focuses on the core of the paper – the hybrid modelling approach, which is then followed by an application sample in the

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 298–303, 2013.

BIVEE project and concluded in section 4 with a summary and pointers toward communities related to the introduced approach.).

2 Interoperability and Conceptual Background

First part of this section is based on the findings of the enterprise interoperability survey from the project ENSEMBLE [7] within the FInES cluster activities. ENSEMBLE [7] observed the different aspects of interoperability. For a detailed discussion on the different aspects we refer to the original document, but would like to highlight different layers and viewpoints of EI as relevant requirements for concept modelling. In [7] ENSEMBLE depicts four levels of interoperability. Level 1 focuses on e.g. Data, Cultural, Software, etc., interoperability. Level 2 for example considers Knowledge interoperability dealing with different semantic abstraction of content to enable human or machine based interpretation whereas Level 3 aims at Cloud interoperability and Level 4 has focus on Ecosystems interoperability in the global world. It is clearly stated that, although maturity level of aforementioned interoperability aspects have been evolved to elaborate them individually, still there are strong dependencies, overlapping and similar common issues among them. . Major requirement is therefore that different concept models may support different dimensions of EI. But different concept models for different dimensions of EI must be on a holistic platform that shares those parts that are in common and enables a specialisation of those parts that are dimension specific. The term “model” is interpreted with the meaning discussed in [8], where a model is “a representation of either reality or vision” [15], that are created “for some certain purpose” [13] “with an intended goal in mind” [27]. Hence the benefit of models can be described in four types: (1) to act as a clear specification, reduce complexity, allow a structured approach and, due to a common understanding, support a participative creation. ([9], [10]); (2) to target semi-automatic implementation of software like Model Driven Architecture; (3) to support knowledge management or (4) to evaluate current status against modelled target goals. Hence, targeting aforementioned EI with concept models, means that pre-defined diagrammatic concepts are available that have a specific meaning enabling to reconstruct relevant parts of the reality in order to either (1) specify, (2) support execution, (3) represent knowledge or (4) evaluate the different dimensions of EI. According to the framework described in [11] and [12], the building blocks of a modelling method include: (1) the modelling language that is most prominently associated with concept models, as available concepts to be used for such model are pre-defined according their semantic, their syntax and their graphical notation, (2) the modelling procedure which defines the stepwise usage of the modelling language and hence is not always available and (3) mechanisms and algorithms that enable the computer-based processing of models. So we can state that all concept model approaches can be described with the aforementioned framework and vice versa, every new concept model approach needs to reflect firstly how to realise the relevant building block. Meta-modelling can be used as a realisation approach to develop domain-specific modelling tools and hence enable IT-supported concept modelling as described previously. The modelling language is understood as the meta-model, which for example is defined in a meta model language like ALL (ADOxx Library Language [5]). The specification of the meta-model can be again defined by a model – the so called meta meta model. Meta Model approaches have been analysed in [14] and can be distinguished in (i) domain, (ii) design – macro and micro level – as well as (iii) integration.. Comparing aforementioned list of identified meta model approaches in the literature it is obvious that most of the interoperability issues raised in the beginning of this section are already covered by individual meta-model approaches. Hence, there are meta models suitable to cover certain aspects of EI dimensions, which can be specified with the generic modelling framework and derived from a meta meta model.

3 Hybrid Modelling as Solution

Hybrid modelling is a concept that merges several meta models and hence applies a similar approach in meta modelling that service orientation applies in software engineering. In the first step the different meta models are classified according (a) their domain, (b) the level of technical granularity, (c) the degree of formalisation and finally (d) the cultural dependency of the applying community. Second, the so called meta model merging is applied. It is seen as the composition of such meta models, and distinguishes in (a) loose integration of meta models, (b) strong integration and (c) hybrid integration. In the PhD thesis of Kühn [12], meta model merging patterns ranging from loosely coupled to tightly coupled meta models are introduced. Loose coupling is very flexible, whereas tight coupling enables the realisation of additional functionality. Meta modelling merging patterns can be, based on [16], summarised as: (1) *Reference pattern*, where two meta-models are complementary and should not or cannot be changed. (2) *Transformation pattern*, where two meta-models are in principle complementary but part of one meta model correspond or can be created out of parts from the target meta model. (3) *Use or aggregation pattern* uses part of the meta model in another meta model or aggregate them into a new meta model, (4) *Merge and extension patterns* of meta models are applied if meta models are closely related and cover similar issues. The challenge is to develop a new or extended meta-model that still keeps the same concepts of the original meta model. (5) *Semantic Lifting* of meta-models is a special way of meta model merging, where a domain specific meta model is merged with the semantic meta model – in most cases an ontology. Research in semantic lifting in the domain of EI is one of the activites within the BIVEE [3] project as outlined in section 3.3.

3.1 ADOxx Technology for Hybrid Modelling Solutions

This section provides an overview on the key points of the ADOxx.org [5] development platform, to enable the discussion how hybrid modelling can be realised. Hence this section refers to the public tutorials on ADOxx.org, where the development platform and the corresponding tutorials can be downloaded. ADOxx® platform provides two pre-defined meta-models that can be used to build domain-specific meta-models. The first one depicts the operative semantic of directed graphs to represent dynamic aspects, and the second focuses on the operative semantic of tree-like structures in the organizational context to represent static aspects. Hence, the dynamic meta-model provides start, activity, decision, parallelity, merging and the end concept. Additionally there are two classes with container semantic, the aggregation and the swim-lane that automatically groups the elements drawn inside them. The static meta-model realizes an organisational structure with persons, resources as well as with above mentioned containers. A new meta-model is developed, through inheriting from aforementioned pre-defined and defining new classes. ADOxx® platform provides further functionality to realize individual mechanisms or algorithms. Generic functionality is provided on root classes of the meta-model for (a) modelling, (b) querying, (c) transformation and (d) simulation. Generic functionality can be extended by a script language called AdoScript that provide more than 400 APIs to interact with the platform. Functionality can be added either by implementing it within the tool or from outside the application using batch files or Web Service access to invoke AdoScripts.

3.2 Technical Solution for Hybrid Modelling on ADOxx

Strong and intermediate integration of meta-models is performed before the modeling tools are deployed, whereas loose integration enables deployment of modelling tools before actual integration is carried out. *Strong and intermediate Integration of Meta Models* is realized by merging two meta models, extending one meta model with another meta model, using part of one meta model in another meta model or aggregating concepts from two meta model into one new concept of the new meta model. It is supported by using text editors with syntax highlighting using the meta model language ALL, where parts of or the whole meta model can be merged, extended, used or aggregated into another meta model. The user should not be aware of the fact that actually two meta models are used. The separation of concern is realized by model types. The graphical notation of meta models enable different views, so the user is guided by the tool menu. This enables to share objects or reference objects with so-called "interref" (a functionality to point to another object), hence the user may access features of any modelling tool. *Loose Integration of Meta Models* such as semantic lifting is a special form of referenced meta model merging. Depending on the level of the user friendliness and applicability, there are different ways to achieve the loose integration. Strict loose integration has no changes in the meta model at all, but it lacks of user friendliness. User friendly solutions require adaptations in the meta model. There are different implementation variants available such as: (1) *Non supported direct linkage* requires no changes in the meta models, the user needs to manually enter the linkage in an existing suitable attribute*;* (2) *Supported direct linkage*, can be realized by an AdoScript that accesses the other modelling tool and enables the selection of an object*;* (3) *Indirect linkage* can be realized using a so-called transit model type where concepts of the corresponding other meta model are included. Hence user friendly mechanisms to reference a model objects (e.g. an interref) can be used. This results in redundant data storage; hence the redundancy must be managed. (4) *Loose coupling* is a special form of indirect linkage, as the concepts that are referenced too are not the target concepts but a reference ontology, which is referenced by both the source and the target concept*; (5) Direct and indirect linkage* is a combination of supported and non-supported linkage, by supporting a fixed core set of concepts but permit the flexibility to also allow agile evolving concepts. All aforementioned solutions have their advantages and disadvantages when supporting the end user – easier one may be too error prone (e.g. free text linkage), and other may involve more resources to be implemented or be too strict (e.g. fixed set of concepts).

3.3 Hybrid Modelling in BIVEE

Hybrid modelling approach has been realized in the BIVEE project. The setting was concerned with the improvement of the Innovation processes in Virtual Enterprises. It was applied to integrate the meta models of the Virtual Innovation Space (VIS) and the traditional Value Production Spaces (VPS) of a Virtual Enterprise (see [17] for details). The selected approach was the realization scenario of the semantic lifting with the semantic meta model of the Production and Innovation Knowledge Repository (PIKR) [18]. VPS meta model itself is deducted based on different aspects including: (a) The Supply Chain Operations Reference [19]. (b) The Value Reference Model [20], (c) SixSigma [24], (d) e3value Model [22], (e) CIMOSA [23], Zachmann [25] and TOGAF [21]. Additionally to these publicly available aspects, a survey on meta-models for production and logistics has been reported and taken into account for VPS within ComVantage research project [26]. These aspects have been reflected in the so-called modelling stack of the VPS meta model comprised of the Value Production Space Chart, Product, Process, Network, KPI, IT-System Pool, and the Artifact Pool. These models, due to necessity to enhance the user friendliness have been integrated following the Strong integration pattern. The dynamic part was achieved by adding the Semantic Transit Model (STM) to the VPS meta model and enabling the semantic lifting based integration with the VIS. The semantic lifting of the VPS meta-model, applies a loose coupling through a transit model (STM) with the PIKR and enables application of the VIS concepts in the model and vice versa.

4 Conclusion and Outlook

The paper started with the observation on EI, their dimensions and their corresponding requirements on model-based approaches. Concept modelling has been introduces as a mature field of knowledge representation that is most commonly realized using meta-model as technology. Hybrid modelling as an instrument has been introduced to show how complex systems can be described. The application of the ADOxx® as an open use meta-modelling platform has been described in order to explain how hybrid modelling approaches from theory, can be implemented into a software platform. The EU-project BIVEE was used to demonstrate how hybrid modelling has been realized in the so-called Value Production Space. An approach presented in the paper is a research topic at the laboratory of the Open Model Initiative [4] and technology to implement the hybrid modelling is provided through the open development platform ADOxx.org [5].

References

1. Fraunhofer, `http://www.iao.fraunhofer.de/lang-de/geschaeftsfelder/unternehmensentwicklung-und-arbeitsgestaltung/1009-industrie-40.html` (access: February 28, 2012)

2. FINES Cluster, FINES Research Roadmap, http://www.fines-cluster.eu (access: February 28, 2012)
3. BIVEE project, http://www.bivee.eu/ (access: February 28, 2012)
4. Open Models Initiative Laboratory, http://www.omilab.org (access: February 28, 2012)
5. ADOxx, http://www.adoxx.org (access: February 28, 2012)
6. Ensemble EU-Project (concluded in 2012.08 but redirected to FINES Cluster), http://www.ensemble-csa.eu (access: February 28, 2012)
7. Ensemble EU-Project, D2.1 Enterprise Interoperability Scientific Base, State of Play Report, http://www.fines-cluster.eu (access: February 28, 2012)
8. Karagiannis, D., Grossmann, W., Höfferer, P.: Open Model Initiative: A Feasibility Study, http://www.openmodels.at (access: February 28, 2012)
9. Whitman, L., Huff, B., Presley, A.: Structured models and dynamic systems analysis: the integration of the IDEF0/IDEF3 modeling methods and discrete event simulation. In: IEEE 29th Conference on Winter Simulation (1997)
10. Davenport, T.H.: Process Management for Knowledge Worker. In: Handbook on Business Process Management, vol. 1. Springer (2010)
11. Karagiannis, D., Kühn, H.: Metamodelling Platforms. In: Bauknecht, K., Tjoa, A.M., Quirchmayr, G. (eds.) EC-Web 2002. LNCS, vol. 2455, p. 182. Springer, Heidelberg (2002)
12. Kühn, H.: Methodenintegration im Business Engineering, PhD Thesis (2004)
13. Object Management Group, http://www.omg.org (access: March 25, 2013)
14. Karagiannis, D., Höfferer, P.: Metamodels in Action: an Overview – First International Conference on Software and Data Technologies: IS27-36. Insticc Press, Setúbal
15. Whitten, J.L., Bentley, L.D., Dittman, K.C.: Systems analysis and design methods, 6th edn. McGraw-Hill Irwin, Boston (2004)
16. Kühn, H., Bayer, F., Junginger, S., Karagiannis, D.: Enterprise Model Integration. In: Bauknecht, K., Tjoa, A.M., Quirchmayr, G. (eds.) EC-Web 2003. LNCS, vol. 2738, pp. 379–392. Springer, Heidelberg (2003)
17. BIVEE, D3.1, State of the Art and Mission Control Room, http://www.bivee.eu
18. BIVEE, Deliverable D5.1, Production and Innovation Knowledge Repository, http://www.bivee.eu (accessed: March 25, 2013)
19. Supply Chain Council, http://supply-chain.org/scor/ (access: March 25, 2013)
20. Value Chain Group, http://www.value-chain.org(access: March 25, 2013)
21. The Open Group, http://www.opengroup.org/togaf/ (access: March 25, 2013)
22. e3 Value, http://e3value.few.vu.nl/e3family/e3value/ (access: March 25, 2013)
23. CIMOSA, http://www.cimosa.de/ (access: March 25, 2013)
24. Rother, M., Shook, J.: Learning to See: Value-stream mapping to create value and eliminate MUDA, Lean Enterprise Institute (1999)
25. Zachman Institute for Framework Architecture, http://www.zachman.com/about-the-zachman-framework (access: March 25, 2013)
26. ComVantage Project, D3.1.1, Specification of Modelling Method Including Conceptualization Outline, http://www.comvantage.eu (access: March 25, 2013)
27. Bézivin, J., Gerbé, O.: Towards a Precise Definition of the OMG/MDA Framework. In: Automated Software Engineering, vol. 273. IEEE Computer Society, Washington, DC (2001)

Re-engineering Data with 4D Ontologies and Graph Databases

Sergio de Cesare[1], George Foy[1], and Chris Partridge[1,2]

[1] Department of Information Systems and Computing,
Brunel University, Uxbridge, UB8 3PH, U.K.
{firstname.lastname}@brunel.ac.uk
[2] BORO Solutions Ltd. London, U.K.
partridgec@borogroup.co.uk

Abstract. The amount of data that is being made available on the Web is increasing. This provides business organisations with the opportunity to acquire large datasets in order to offer novel information services or to better market existing products and services. Much of this data is now publicly available (e.g., thanks to initiatives such as Open Government Data). The challenge from a corporate perspective is to make sense of the third party data and transform it so that it can more easily integrate with their existing corporate data or with datasets with a different provenance. This paper presents research-in-progress aimed at semantically transforming raw data on U.K. registered companies. The approach adopted is based on BORO (a 4D foundational ontology and re-engineering method) and the target technological platform is Neo4J (a graph database). The primary challenges encountered are (1) re-engineering the raw data into a 4D ontology and (2) representing the 4D ontology into a graph database. The paper will discuss such challenges and explain the transformation process that is currently being adopted.

Keywords: 4D ontology, perdurantism, foundational ontology, semantic transformation, graph databases, Neo4J, Big Data.

1 Introduction

The amount of data that is currently made available on the Web is growing thanks primarily to the Linked Open Data (LOD) initiative. While LOD data tends to be in formats such as the Resource Description Framework (RDF) and Microformats, there is also an enormous amount of data made available in other structured formats and more often even semi-structured or unstructured. This is the case of data that is privately sold (e.g., by credit risk companies) or made publicly available by Government Agencies (e.g., open government data initiatives) [1].

In order to more effectively process and integrate data from a multitude of sources as well as make it semantically consistent with the existing enterprise data architecture, we have chosen to adopt ontologies. More specifically we adopt a 4D foundational ontology [2, 3] to drive the interpretation of such data sources, improve

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 304–316, 2013.

the semantic expressiveness of the data and harmonise it in a consistent manner. The approach that we are adopting, therefore, begins with the raw data, ontologically interprets and transforms the data in order to extract its semantics and express such semantics in 4D ontologies. These 4D ontologies are then mapped to a graph-based data architecture. Neo4J [4] is the implementation technology that we have currently chosen to adopt. The main reasons for adopting a graph database to persist the ontological models are: (1) the flexibility that a graph structure provides in implementing any modelling paradigm and (2) the scalability it provides in terms of organising and accessing massive amounts of data.

The paper is organised as follows. Section 2 defines the problem in more detail. It also explains the reasons for underpinning the data re-engineering with a foundational ontology and why a graph database was chosen as the implementation technology. Section 3 provides an overview of the 4D foundational ontology adopted to underpin the re-engineering effort. Section 4 explains the challenges of mapping the foundational ontology to a graph and Section 5 presents a few mapping patterns discovered to date. Section 6 describes related work and Section 7 concludes the paper and discusses future work.

2 The Research Problem

This research investigates the problem of integrating large datasets from different sources into one common data repository or into an existing corporate database. While this research focuses on large datasets acquired externally, it must be noted that the approach described in this paper can also be applied by organisations to examine their own vast transactional datasets from which to glean potential competitive information.

The datasets that are being referred to here are, for example, those made available by authorities such as Companies House (the U.K. Company Registration Office). These datasets come in a variety of formats. For example, datasets available at data.gov.uk are currently provided as CSV (Comma-Separated Values) or JSON (JavaScript Object Notation) files. In essence the underlying original structure is that of a spreadsheet. Normally such files and their corresponding JSON representations are direct format conversions from legacy spreadsheets or flat files. This is normally apparent from the denormalised form that such data assumes. While syntactically structured (in terms of rows and columns), much of the semantics of these datasets is implicit and cannot be readily integrated with other datasets. The integration of multiple datasets is not a mere technical problem, but it also represents a business opportunity for organisations to exploit in this new era of the Digital Economy [5]. As stated by the U.K. Cabinet Office, the aim is to create “an information marketplace for entrepreneurs and businesses; releasing valuable raw data from real-time transport information to weather data” [6].

The overall research problem also has another aspect to it, which is performance. In fact, since the amount of data processed can easily be in a range that runs from hundreds of gigabytes to tera/petabytes, there is also a technical challenge of processing so called Big Data [7]. This paper however will only focus on the problem of semantic transformation, which represents the part of the research carried out to date. Future work, as documented in Section 7, will explore the other aspects of the research.

Re-engineering data can be viewed as essentially a problem of semantic interpretation, in other words a process of interpreting the raw data and identifying the things that the data refers to in the real world (or any possible world). This realist approach is greatly simplified by the adoption of a foundational ontology to drive the re-engineering. A foundational (or upper-level) ontology defines the kinds of existence that things can have (i.e., a categorical theory). Categorical theories are studied in Philosophy.

In Philosophy, Ontology, as a discipline, is the study of existence and of the kinds of things that (can) exist. One aspect of existence is change over time, and in this area there are two predominant ontological theories: endurantism and perdurantism [8]. In endurantism a three-dimensional object is wholly present at any given instant and persists by 'sweeping' through a region of space-time (in the words of Sider [8]). Another aspect of ontology is identity; and a key question is whether there is any criterion of identity and what it is. One endurantist approach is to say that while wholly present at all moments of its existence, an object preserves its identity via a set of essential attributes (for example, a person's DNA). The perdurantist approach sees an object as a four-dimensional extension (or extent) in the universe (i.e., occupying a region of space-time) and it is not therefore totally present at any given instant, but instead only partially present. A common perdurantist criterion of identity is the object's four-dimensional extension. In its lifetime an object goes through states (or stages). For example, a person goes through the stages of childhood and adulthood. In perdurantism change is explained via successive temporal parts. Therefore, while an endurantist object persists in three-dimensional space and entirely shifts from one point in time to the next, in perdurantism an object exists in four-dimensional space-time and can be regarded as partially present at any time or portion of its spatiotemporal extension.

In this research we adopt BORO, a 4D foundational ontology, described in the following section. The adoption of a perdurantist ontology is motivated by it being particularly suitable to model the enterprise context and its continuously changing nature. Perdurantism models change by representing stages of a particular object as temporal parts (examples include changes of address, changes of legal status and changes of a company's primary type of activity). Perdurantism and extensionalism naturally allow to model particular objects with intersecting spatiotemporal extents, for example, between a person (*Bill* Gates) and a company position (*CEO of Microsoft*). These aspects of 4D ontologies (along with others) provide more explicit and accurate representations of change in terms of a succession of different temporal parts. Greater accuracy in the models produced can lead to greater levels of flexibility and reusability when evolving information systems (IS) as more thoroughly explained by Partridge [2].

In order for ontologies to provide concrete and visible benefits to IS engineering it is essential to take the ontological models beyond the modelling/design stage and attempt to use them not only to influence the implementation of technological artefacts (e.g., databases and software), but preferably to realise the ontologies in the technology itself. This means being able to take a foundational ontology with the modelled domain ontologies and create a database or software implementation that maintains high levels of direct traceability to the ontology. With most traditional paradigms (e.g., relational databases and object-oriented languages), aligning the technological implementation to the ontology is possible, but given the paradigm mismatch, development normally occurs with 'workarounds' that may have a negative impact on ontological alignment.

In graph databases [9] representations assume a graph form with nodes and edges. Edges represent relations between nodes. Properties can be defined for both the nodes and the edges. Graph databases are schemaless; this means that, unlike relational databases in which data must be represented and stored in a rigid structure with tables, rows and columns, the only structural constraint that graph databases dictate is the graph structure (a network of nodes connected by edges). This allows the modeller/developer significantly increased flexibility in the way the data is represented. From a metamodelling perspective this implies that the metamodel can be treated as data and represented as a graph and combined with its model instantiations also represented in the same graph. In our case the foundational ontology represents the metamodel and the domain ontologies represent the metamodel instantiations. As a consequence, our working research 'hypothesis' is that a schemaless database would enable us to implement the database in a form that more closely resembles the 4D ontology.

3 A Perdurantist Foundational Ontology

BORO, developed by Partridge [2], is a perdurantist upper level ontology strongly based on extensionality. BORO influenced the ISO 15926 standard and inspired the upper level ontology of the International Defence Enterprise Architecture Specification for exchange Group [10], adopted by the U.S. Department of Defense Architecture Framework (DoDAF). BORO has been applied in various industrial sectors including finance, oil and gas, and defence.

The aim of this section is to present the BORO foundational ontology and provide the reader with the fundamental knowledge to understand the work described in the remainder of the paper. It is beyond the scope of this paper to provide an exhaustive explanation and definitions of the whole foundational ontology. For an in depth presentation of BORO the reader is invited to refer to Partridge [2] in its original form or IDEAS [10] for a slightly modified, yet still detailed, version.

From a philosophical perspective the BORO foundational ontology explicitly addresses a set of metaphysical choices. BORO has adopted: (1) a realist stance towards ontology, that is it takes for granted a mind-independent real world; (2) a revisionary stance – accepting that if we want better models, we need to change the ways we look at the world; (3) completeness categories based upon extensional criteria of identity and (4) a 4D and possible worlds approach as these fit best with its commitment to extensionalism [11].

Figure 1 presents a graphical representation of the foundational ontology. The names of the foundational objects are prefixed with 'F_'. The notation is that of the Unified Modelling Language (UML).

At its highest level the BORO foundational ontology represents:

- *Objects*: Anything that exists. (In IDEAS the term *Thing* is used in place of *Objects*.)
- *Elements*: An element is a physical body with a spatiotemporal extent (i.e., particulars).
- *Types*: A type is a set or class of objects (i.e., universals). The extension of a type is given by all the objects of that type. Objects of a certain type are said to be instances of that type. Types can have individual instances (*ElementTypes*), type instances (*Powertypes*) or tuple instances (*TupleTypes*). Only *TupleTypes* are explicitly represented in Figure 1.
- *tuples*: A tuple is a relationship between two (in the case of *couples*) or more objects. Examples of subtypes of *tuples* include *typeInstances*, *superSubTypes*, *powertypeInstances* and *wholeParts*.
- *TupleTypes*: A type whose instances are tuples. There is a *powertypeInstance* relation between *TupleTypes* and *Tuples*.
- *TemporalParts*: A temporal part is an individual whose spatiotemporal extent is part of another individual.
- *Events*: An event is an individual temporal part that does not persist through time (i.e., an event has zero 'thickness' along the time dimension). Events represent temporal boundaries that either create (*CreationEvents*) or dissolve (*DissolutionEvents*) individuals (e.g., a person) or individual temporal parts that persist through time (i.e., states).
- *States*: A state is a temporal part of an individual that persists through time. States (and elements in general) are bounded by events. A state can have further temporal parts (i.e., states and events).
- *happensTo*: This tuple type relates an event with one or more individuals affected by the event. *happensTo* has two subtypes:
  - *creates*: Relates a creation event with the element(s) whose creation is triggered by the event.
  - *dissolves*: Relates a dissolution event with the element(s) whose dissolution is triggered by the event.
- *happensAt*: This tuple type relates an event with a time instant or interval (*TimeInstantsOrIntervals*) and it indicates the time at which an event takes place.
- *temporalPartOf*: This tuple type relates an individual with its temporal parts (states and/or events).

To visually clarify how BORO as a perdurantist ontology models the real world including change, let us consider a simple example of a company (*Acme Company Ltd.*) who during the course of its life changes its primary business activity from the production of paper to the production of mobile phones. Such information is normally legally required by Company Registration Offices. This is represented in Figure 2. As the figure shows *Acme* (as a 4D element) extends through space-time. A portion of *Acme*'s extension has a temporal part named '*Business Activity 1*' representing the company's paper manufacturing stage (or state) and another temporal part named

'*Business Activity 2'* representing the mobile phone manufacturing stage. Both stages have extents that are physically part of *Acme*'s overall spatiotemporal extension. There are also three events implicitly represented by the lines that bound the states. The temporal boundary on the left of '*Business Activity 1'* represents the event creating that stage, the boundary lying between '*Business Activity 1'* and '*Business Activity 2'* represents an event that dissolves the first state and creates the second state. The boundary to the right of '*Business Activity 2'* dissolves this state.

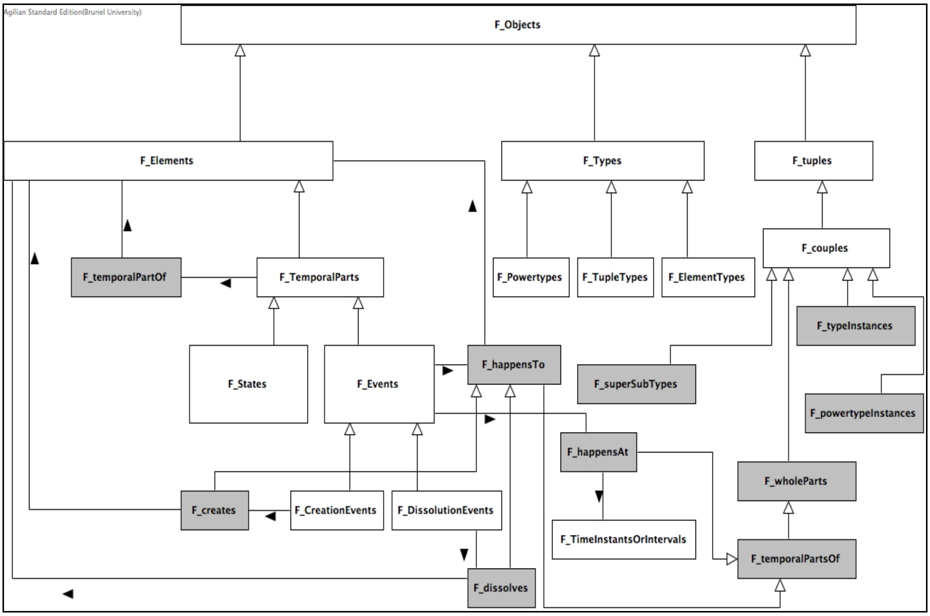


Fig. 1. BORO foundational ontology (partial view). (*TupleTypes* are represented in light grey).

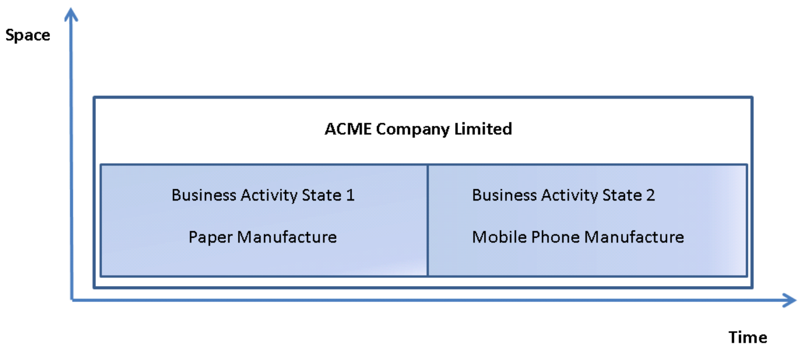


Fig. 2. Example space-time map

4 Implementing the Upper Ontology

4.1 The Foundational Graph

In order to represent ontological models in a database (in our case a graph database) it is necessary to represent and load the foundational ontology first of all. It is the foundational ontology that enables the representation of the domain ontology derived from the raw data. In essence things in the domain ontology will instantiate or subtype the high-level types of the upper level (i.e., the types represented in Figure 1). Since all models are a graph, the model in Figure 1 should ideally be transposed as it stands into the graph database. However a few considerations must be made.

In BORO *tupleTypes* and *tuples* are first-class objects. As Figure 1 shows *tupleTypes*, like *F_happensTo* and *F_creates*, are not simply drawn with the UML association notation but explicitly represented with the UML class symbol. This is necessary in order to allow the ontologist to describe the *tupleTypes* and *tuples* themselves. For example, subtyping a *tupleType* as in the case of *F_temporalPartsOf* and *F_wholeParts*.

In the graph implementation we have therefore chosen to maintain the same explicit representation of *tupleTypes* and *tuples*. Therefore when representing a relation (*R1*) between two things (e.g., *Prince William* and *Prince Charles*) in Neo4J, the relation is not simply represented with two nodes and an edge (i.e., [William → Charles]), but with three nodes and two edges. This enables us to say the following:

[*William* → *R1* → *Charles*] (1)
[*childOf* → *typeInstances* → *R1*] (2)
[*tuples* → *superSubTypes* → *childOf*] (3)
[*tupleTypes* → *typeInstances* → *childOf*] (4)

While it is important to explicitly represent relations, there are three upper-level relations that are unsurprisingly very widely used: *typeInstances*, *superSubTypes* and *powertypeInstances*. In these three cases we have decided to simply name the edges (as shown in the above listing) rather than reify the relations. The name would be implemented as a property of the edge. This makes the graph more compact with, in this context, losing required explanatory power. Otherwise a relation like (3) would become:

[*tuples* → *R2* → *childOf*] (5)
[*superSubTypes* → *typeInstances* → R2] (6)

With *typeInstances* relations there is another reason for using this compact form. The problem in Philosophy is known as the Third Man Argument (or Bradleyian Regress) and leads to an infinite regress of reified relations. For example, in (5) of the above listing the reification of the typeInstances relation leads to the following:

[*tuples* → *R2* → *childOf*] (7)
[*superSubTypes* → *R3* → *R2*] (8)
[*typeInstances* → *R4* → *R3*] (9)
[*typeInstances* → *R5* → *R4*] (10)
ad infinitum …

R3, *R4* and *R5* are all instances of *typeInstances* leading to an infinite chain of relations.

4.2 Graphs of Domain Patterns

Besides being a foundational ontology, BORO also defines a method for discovering general ontological patterns from existing systems and data. These general patterns enforce reusability and enable the ontologist to apply existing semantic models to the knowledge discovered from the interpreted data. As with the foundational ontology, these general patterns must also be loaded into the graph database before they can be used; however unlike the foundational layer, such patterns can be loaded in parallel with the semantic interpretation of the data as long as they are present in the database prior to their use.

An example of a general pattern is the Naming Pattern (in the model the prefix used is 'N_') represented in graph form in Figure 2. The *N_Names* type represents the set of all possible names in the world. *N_Names* is a type and not an element. For example, the name *John* is considered as the set of all utterances (written, oral, etc.) that name people called *"John"*. A naming space is a set of names; for example, the set of registration numbers that Companies House issues to uniquely identify a company.

It is important to note that while all nodes of the graph are labelled by a 'name' property in Neo4J, this property merely names the modelling element and it is not meant to be a name for the thing being modelled. The only exception is the name

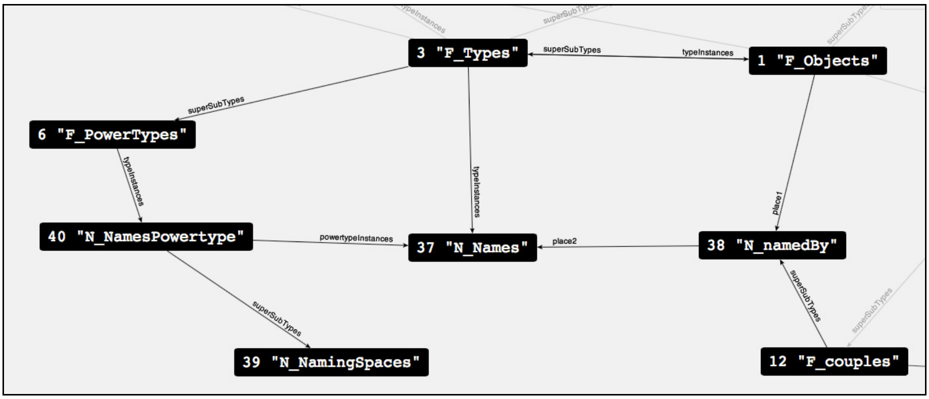


Fig. 3. Naming Pattern in Neo4J

property of *N_Names*. Names of real world objects must be represented as instances of the *N_Names* type to which the named object is related via the N_*namedBy* tupleType.

For reasons of space other patterns are not shown here but will be referred to later in the paper.

5 Semantic Transformation of the Data

This research uses a dataset acquired from Companies House in the U.K. The dataset was provided as a CSV file of approximately 3 GB corresponding to a spreadsheet of 173 columns and approximately 3.5 million rows. An extract of the column headings is provided in Table 1. The data was managed with custom-built code written in Python importing the standard CSV module. For the creation of the graph database the py2neo API was used in order to send REST requests to the Neo4J server.

Table 1. Extract of the column headings of the data file

registration_ number	legal_status_ code	date_inc	latest_accounts_date	latest_ar_date

After implementing the foundational ontology and those patterns deemed relevant to the domain (e.g. names, persons and intentionally constructed objects), the semantic interpretation of the data proceeded as follows.

First, the set of rows was interpreted. This implies understanding what a row refers to and the type it is an instance of. In this case each row refers to an individual U.K. company. The type instantiated is named *UKCompanies*. Once the meanings of the elements and the type have been determined, these objects must then be related to existing patterns and via the patterns to the foundational ontology. If no existing pattern appears to be relevant then this may be a sign that a new pattern may be hidden in the data and possibly discovered through further analysis. In this instance the Persons pattern previously loaded can be reused. In fact *UKCompanies* is a subtype of *LegalPersons* which in turn is a subtype of *Persons*.

The next (and most significant) step is to iterate through the columns of the spreadsheet and semantically interpret them. While the interpretation of the set of rows was relatively straightforward (at least in our case), interpreting column data presents some interesting challenges. There are a few mapping patterns (MP) that have emerged and summarised as follows:

MP1: If r_i represents the specific element that a row refers to and R_i its type, then one can think of a column as representing a type (C_i) and the intersection with the row (i.e., each cell) explicitly representing an instance of C_i (or c_i). Implicitly represented are also a tuple type (T_i) and a tuple (t_i). For example, with the first column (named "registration_number") the mapping in Table 2 emerges.

Table 2. Example of Mapping Pattern MP1

Spreadsheet Type	Value	Refers to	Referent	BORO Foundation
Column name (C_i)	registration_number	Set of all registration numbers assigned by Companies House	CHRegNumbers	F_Types
Cell value (c_i)	"0000006"	Individual registration number assigned to a company	"0000006" (instance of CHRegNumbers)	F_Elements
Implicit relation (T_i)	n/a	Set of all relations between UK Companies and Registration Numbers	namedByCHReg-Number	F_TupleTypes
Implicit relation (t_i)	n/a	Relation between a specific UK company and the registration number "0000006"	The tuple: (Company6, 0000006)	F_tuples

MP2: In many cases the columns cannot be interpreted in isolation because their values represent elements that have relations between them. For example, there are columns representing the different parts of a company's address (street and number, city, county, etc.). In this case there exists a *wholeParts* relation between them respectively. As a consequence the rules in MP1 are applied along with another interpretation rule which maps (and makes explicit) the implicit relations between the types represented by the columns and the pairs of elements represented by the cell values of those columns on the same row.

MP3: Some cells contain values that encode and map to more than one real world element or even to an entire classification structure. An example of the former is a composite address (e.g., 123 Main Street). In this case '123' refers to a specific building while 'Main Street' refers to a whole street. The latter case (which actually may be a mapping pattern in its own right) is exemplified by the U.K. Standard Industry Codes (SIC). In this case a specific code, expressed as a string (of numeric characters) in the spreadsheet, codifies a structure in which the code can be broken down into parts, with each part representing successive groupings of companies. In SIC terminology these groupings are called sector, division, group, class and subclass. Coding schemes like SIC are classification systems, which BORO can handle well with *PowerTypes* (or the set of all subtypes of a given type) in conjunction with *superSubTypes* and *powertypeInstances* (two tuple types of the foundational ontology). This type of semantic transformation allows us to explicitly model and refer to an entire classification system (i.e., SIC), relate it to other classification systems (for example, successive versions of SIC) and relate it to a naming space (in this case U.K. SIC codes). This is an effective example of how BORO is capable of transforming a set of simple codes (e.g., "0311") into a complex ontological structure. Thanks to BORO's strong extensionality principle a clear and explicit distinction is made between a classification system and its naming space.

MP4: This mapping pattern builds on MP3 and relates to cases in which there is a succession of columns that represent a type of change that a company may undergo in its lifetime. Examples include a change of address, change of name, change of SIC code, etc. While there are subpatterns that are not discussed here for limitations of space, a typical case is one in which there are columns representing the current status (e.g., current address) and the previous four statuses (addresses) plus further columns specifying when the various changes (of address) occurred. The mapping to a 4D ontology translates into a new subtype of *F_States* (e.g. *CompanyX_at_AddressY*), a new subtype of *F_Events* (e.g., *AddressChange*) and all related subtuples/types that this entails as modelled in the foundational ontology. While the original data was limited to the representation of only four previous addresses, the 4D graph model can record an unlimited number of changes. This pattern is a clear example of the effectiveness of the 4D approach in modelling change. In fact, with each change of address the new state represents a temporal part of the company, which can be itself related to its corresponding address, thus providing a more objective model of what occurs in the real world.

The above mapping patterns are a non-exhaustive list. As the research progresses and more data is semantically analysed, we expect to discover further mapping patterns with more clearly defined rules, as well as refine the existing rules. At this stage the mapping is being carried out manually and the translation rules are being encoded in Python on a case-by-case basis. We envisage that once these patterns are tested against a greater amount of data, we will be able to develop generic APIs for each consolidated and tested mapping pattern so as to gradually proceed to a much higher level of automation for the further 4D re-engineering of data.

6 Related Work

The focus of the work presented in this paper is on the semantic transformation of large amounts of data (e.g., Big Data) acquired typically from heterogeneous sources and in semi-structured raw formats (e.g., CSV files). This problem is part of a broader research area related to the re-engineering of data and systems. In fact the challenges encountered and described here are typical of most projects that adopt processes similar to what is known as Extraction-Transformation-Loading (ETL) [12]. While ETL is mostly applied in Data Warehousing, the general problems are common. In our case, however, the main differences lie in the transformation phase, which is entirely driven by a 4D foundational ontology. In brief, for us transformation consists of Semantic Interpretation (translation into the 4D paradigm), Semantic Improvement (generalise to existing patterns or identify general reusable ontological patterns), and Semantic Harmonisation (consistently integrate the new ontologies and patterns into the existing ontological graph). Moreover previous work carried out on ontology-driven ETL (e.g., [13]) normally adopts a Semantic Web (OWL/RDF) approach rather than be driven by a philosophically grounded foundational ontology.

Further literature (for example, see [14] and [15]) also investigates ontologies in the context of public data and its use for providing information services. However, even here, as with most ontology related ETL work, the focus is on Semantic Web Technologies (primarily Linked Data) with no grounding in philosophical foundational ontologies.

An example of previous research that has also investigated the use of foundational ontologies to derive domain models and ontology patterns from Web resources is

SmartWeb Integrated Ontology (SWIntO) [16]. However in this case the authors adopted a foundational ontology (merger of DOLCE and SUMO), which focused on linguistic-cognitive aspects. This was fully justified in this project due to SmartWeb system's heavy reliance on linguistic information. In fact one of the main objectives of the research was to "produce domain-specific ontologies that are relevant for mobile and intelligent user interfaces to open-domain question-answering and information services on the Web". In our case, a linguistic-cognitive based ontology would not have been suitable since our aim is for the re-engineered models to ultimately be fully integrated with an organisation's corporate knowledge assets, hence our decision to adopt a foundational ontology that is context-independent and mind-independent. Such an approach appears, in our view, more appropriate toward facilitating the integration of multiple data sources with the existing corporate data.

7 Conclusion

This paper summarised the initial findings of research-in-progress aimed at re-engineering large amounts of raw data with a 4D ontology and implementing the ontology in a graph database (Neo4J). The research thus far uncovered some of the major challenges related to (1) implementing a 4D foundational ontology in a graph database, (2) semantically interpreting raw data in a spreadsheet format to a 4D ontology and (3) identifying mapping patterns which, with further testing, can help to move from a manual data translation into a more automated mapping to the 4D ontology and consequent loading of the semantically interpreted model into Neo4J.

Once the semantic problem of translating the data from its raw form to a 4D ontology and a 4D graph is proven effective our next step will be to integrate further data sets which besides being grounded in the 4D foundational ontology must be harmonised with the previously reengineered models. We will also explore developing an automated solution that will adopt the discovered mapping patterns to transform the data into 4D representations. Performance will also be fundamental for information retrieval and general usage once the system goes into the production stage. This means that while the database will be semantically rich and expressive thanks to its strong ontological foundation, it must also run on an architecture that guarantees high accessibility and performance [7].

Acknowledgement. The authors wish to thank James Dobree, CEO of Level Business Ltd., for funding this research.

References

1. Jain, P., Hitzler, P., Sheth, A.P., Verma, K., Yeh, P.Z.: Ontology Alignment for Linked Open Data. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part I. LNCS, vol. 6496, pp. 402–417. Springer, Heidelberg (2010)
2. Partridge, C.: Business Objects: Re-Engineering for Re-Use. Butterworth-Heinemann (1996)

3. Partridge, C., Mitchell, A., de Cesare, S.: Guidelines for developing ontological architectures in modelling and simulation. In: Tolk, A. (ed.) Ontology, Epistemology, & Teleology for Model. & Simulation. ISRL, vol. 44, pp. 27–57. Springer, Heidelberg (2013)
4. Neo4J, `http://www.neo4j.org`
5. McAfee, A., Brynjolfsson, E.: Big Data: The Management Revolution. Harvard Business Review (October 2012)
6. Minister of State for the Cabinet Office and Paymaster General, Open Data White Paper: Unleashing the Potential, `http://data.gov.uk/sites/default/files/Open_data_White_Paper.pdf` (last accessed on April 03, 2013)
7. Pereira, A.L., Appel, A.P.: Modeling and Storing Complex Network with Graph-Tree. New Trends in Databases and Information Systems. Advances in Intelligent Systems and Computing 185, 305–315 (2013)
8. Sider, T.: Four-Dimensionalism: An Ontology of Persistence and Time. Oxford University Press, USA (2002)
9. Angles, R., Gutierrez, C.: Survey of Graph Database Models. ACM Computing Surveys 40(1) (2008)
10. IDEAS Group: The IDEAS Model, `http://www.ideasgroup.org/foundation/`
11. Partridge, C., Mitchell, A., de Cesare, S.: Guidelines for Developing Ontological Architectures in Modelling and Simulation. In: Tolk, A. (ed.) Ontology, Epistemology, & Teleology for Model. & Simulation. ISRL, vol. 44, pp. 27–57. Springer, Heidelberg (2013)
12. Vassiliadis, P.: A Survey of Extract-Transform-Load Technology. International Journal of Data Warehousing and Mining 5(3), 1–27 (2009)
13. Skoutas, D., Simitsis, A.: Ontology-based Conceptual Design of ETL Processes for both Structured and Semi-structured Data. International Journal on Semantic Web and Information Systems 3(4) (2007)
14. Shadbolt, N., O’Hara, K., Berners-Lee, T., Gibbins, N., Glaser, H., Hall, W., Schraefel, M.C.: Linked Open Government Data: Lessons from Data.gov.uk. IEEE Intelligent Systems (May/June 2012)
15. Alani, H., Dupplaw, D., Sheridan, J., O’Hara, K., Darlington, J., Shadbolt, N.R., Tullo, C.: Unlocking the Potential of Public Sector Information with Semantic Web Technology. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC/ISWC 2007. LNCS, vol. 4825, pp. 708–721. Springer, Heidelberg (2007)
16. Oberle, D., Ankolekar, A., Hitzler, P., Cimiano, P., Sintek, M., Kiesel, M., Mougouie, B., Baumann, S., Vembu, S., Romanelli, M., Buitelaar, P., Engel, R., Sonntag, D., Reithinger, N., Loos, B., Zorn, H.P., Micelli, V., Porzel, R., Schmidt, C., Weiten, M., Burkhardt, F., Zhou, J.: DOLCE ergo SUMO: On foundational and domain models in the SmartWeb Integrated Ontology (SWIntO). Web Semantics: Science, Services and Agents on the World Wide Web 5, 156–174 (2007)

Supporting Customer Choice with Semantic Similarity Search and Explanation*

Anna Formica, Michele Missikoff, Elaheh Pourabbas, and Francesco Taglino

National Research Council, Istituto di Analisi dei Sistemi ed Informatica "A. Ruberti"
Viale Manzoni 30, I-00185 Rome, Italy
{name.surname}@iasi.cnr.it

Abstract. Semantic search and retrieval methods have a great potentiality in helping customers to make choices, since they appear to outperform traditional keyword-based approaches. In this paper, we address *SemSim*, a semantic search method based on the well-known information content approach. *SemSim* has been experimented to be effective in a defined domain, namely the tourism sector. During experimentation, one of the first requests raised from the users concerned the possibility to explain, besides the typical output of a semantic search engine, why a given result was returned. In this paper we investigate *SemSim* with the aim of providing the user with an explanation about the motivations behind the ranked list of returned options, with graphical representations conceived to better visualize the results of the semantic search.

Keywords: Similarity Reasoning, Weighted Reference Ontology, Information content, Digital Resources.

1 Introduction

Similarity reasoning represents a promising application of semantic technologies that is constantly gaining importance in information management. It also represents one of the key pillars of a new generation of search engines, capable of processing a user request from a conceptual perspective instead of basing the research on keyword-matching mechanisms. Along this direction, several semantic approaches have been proposed in the literature, based on the use of reference ontologies. Such approaches require that both the user request and the elements of the search space are semantically annotated, i.e., are associated with semantic expressions built upon the concepts of a reference ontology. This approach, referred to as *semantic* search and retrieval, appears to outperform traditional keyword-based approaches. However, the application of semantic search is restricted to focused domains due to the cost needed to develop the required semantic infrastructures (basically, reference ontology and semantic annotations).

In a previous paper [5] the authors have proposed *SemSim*, a semantic search method based on the idea that the similarity between conceptual structures depends on the amount of information content they share. In turn, the information content carried by a

* This work has been partially supported by the European project BIVEE: Business Innovation and Virtual Enterprise Environment (No. FoF-ICT-2011.7.3-285746).

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 317–328, 2013.

conceptual structure depends on the information content carried by the concepts (features) of the reference ontology (an *ontology* is a formal, explicit specification of a shared conceptualization [7]) for the given domain, at a given time. Intuitively, the information content of a concept, in our approach, depends on the number of objects in the search space that carry such a concept. In brief, a concept associated with fewer objects has a better selection coefficient and therefore a greater information content. Conversely, a concept that can be found in a large number of objects has less selectivity and therefore carries a more limited information content. Then, we introduced the notion of a *weighted ontology*, where each concept is associated with a weight corresponding to the frequency of the search objects that carry such a concept. It is quite obvious that concepts up in the hierarchy will have higher weight (and therefore less information content) than more specific concepts, positioned down in the hierarchy [11].

The core of the *SemSim* method consists in the derivation of the semantic similarity between concept structures, one representing a user request and one representing a target resource. Currently, we are experimenting the method, and the possibility to obtain, besides the typical output of a semantic search engine (i.e., a list of target objects, ranked according to a similarity score), a synthetic explanation of why the similarity engine achieves the displayed results. Therefore, the objective of this work is to extend the *SemSim* method by providing the user with such an explanation about the mechanism adopted in deriving the ranked list of objects. It is worth noting that often the outcome is not intuitive, since *SemSim* aims at maximizing the global information sharing of the user request and target object (i.e., the summation of the common information contents for all the concepts characterizing a given search object) therefore, avoiding 'easy' matching mechanisms.

The paper is organized as follows. After the next section presenting the related work, Section 3 recalls some basic notions of the *SemSim* method. In Section 4, a qualitative analysis of *SemSim* is presented, and in Section 5 two different visualizations of the analyzed results are given. Finally, Section 6 concludes.

2 Related Work

In the large majority of papers proposed in the literature [3,13], semantic search is performed by using WordNet [16], see for instance [8,15]. WordNet (a lexical ontology for the English language) provides, for a given concept (noun), the natural language definition, hypernyms, hyponyms, synonyms, etc, and also a measure of the *frequency* of the concept, by using noun frequencies from the Brown Corpus of American English [6]. We did not adopt the WordNet frequencies for two reasons. Firstly, we deal with specialized domains (e.g., systems engineering, tourism, etc.), requiring specialized domain ontologies. WordNet is a generic lexical ontology (i.e., not focused on a specific domain) that contains only simple terms. In fact, multi-word terms are not reported (e.g., terms such as "private house" or "country resort" are not defined in WordNet). Secondly, there are concepts in WordNet for which the frequency is not given (e.g., "accommodation").

The work presented in [8] shares some analogies with our approach with regard to the need of computing weights without relying on large text corpora. Therefore, they propose a method, referred to as CP/CV, such that each node in the taxonomy is associated with a concept vector, built on the basis of the topology of the ontology and the position of concepts therein. Then, the similarity of concepts is evaluated according to the *cosine* similarity of the related concept vectors. Conversely, in our work the similarity of concepts is conceived to determine the similarity of two concept vectors.

Regarding the methods to compute the similarity between concept vectors, our work proposes a two stages method, firstly computing the pair-wise concept similarity (*consim*), and then deriving the similarity between vectors of concepts (*semsim*). Pair-wise concept similarity is performed according to the information content approach [11]. With regard to the second stage, we adopted a solution inspired by the *maximum weighted matching problem* in bipartite graphs. In the literature the *Dice*, *Jaccard* and *Cosine* [12] methods are often adopted in order to compare vectors of concepts. However, in these methods the matchmaking of two concept vectors is based on their intersection, without considering the position of the concepts in the ontology. According to the *Weighted Sum* approach [2] a fixed value (i.e., 0.5) is assigned to each pair of hierarchically related concepts. Our proposal is based on a more refined semantic matchmaking, since the match of two concepts is performed according to their shared information content, and the vector similarity is based on the optimal concept coupling.

Within the semantic web search approaches supported by ontologies, we selected [1] and [10] as representative works. In the mentioned papers, semantic search methods have been conceived for retrieving the relevant resources based on OWL[1] (Ontology Web Language) ontologies. Basically, the relevant resources are identified by weighting the instances and properties defined in the assertional knowledge (ABox). Conversely, in our approach, weights are assigned to the features which correspond to the concepts of the terminological knowledge (TBox), and the identification of the relevant resources is performed on the basis of the ranked feature vectors.

3 The Semsim Method: An Overview

In [5] two different *SemSim* methods were addressed, namely the *probabilistic* and the *frequency* approaches. In the mentioned paper, we have shown that the *SemSim* method based on the frequency approach has a higher correlation with *Human Judgement* (*HJ*) than some of the most representative approaches defined in the literature. In this section, we recall the frequency approach, which is the focus of our paper. The *Universe of Digital Resources* (UDR) is the totality of digital resources that are semantically annotated with a reference ontology. In our work we address a simplified notion of ontology, *Ont*, consisting of a set of concepts organized according to a ISA hierarchy. In particular, *Ont* is a *taxonomy* defined by the pair $Ont = <C, H>$, where C is a set of concepts and H is a set of ordered pairs of concepts of C such that if $(c_i,c_j) \in H$, then c_j ISA c_i, i.e., c_i is a more general concept with respect to c_j.

Given two concepts c_i, $c_j \in C$, the *least upper bound* of c_i, c_j, $lub(c_i,c_j)$, is always uniquely defined in C (we assume the hierarchy is a lattice). It represents the least abstract concept of the ontology that is a more general with respect to c_i and c_j.

[1] `http://www.w3.org/TR/owl2-primer/`

Consider an ontology $Ont = <C, H>$. A *request feature vector* (*request vector* for short) rv is defined by a set of ontology concepts (the order of the concepts is irrelevant), i.e., $rv = (c_1,...,c_n)$ where $c_i \in C$.

Analogously, given a digital resource $dr_i \in$ UDR, an ontology feature vector ofv_i associated with dr_i is defined by a set of ontology concepts describing the resource as follows: $ofv_i = (c_{i,1},...,c_{i,m})$, where $c_{i,j} \in C, j = 1,...,m$.

A *Weighted Reference Ontology (WRO)* is a pair $WRO = <Ont, w>$, where w is a function defined on C, such that given a concept $c \in C$, $w(c)$ is a rational number in the interval [0,...,1].

In the following, let H^* be the reflexive and transitive closure of H. According to the frequency approach, the weight associated with the concept c in the ontology $Ont = <C, H>$, indicated as $w_f(c)$, is defined as follows:

$$w_f(c) = \frac{|\bigcup\{ofv : (c' \in ofv,\ (c,c') \in H^*)\}|}{|UDR|}$$

Therefore, the weight assignment is computed on the basis of the given ontology feature vectors and weights are assigned according to a bottom-up approach in order to take into account the essence of the ISA hierarchy. For instance, in our running example in the tourism domain, suppose the digital resources are vacation packages for visiting a European capital, which are offered by a tourism agency. Each package is annotated with one ontology feature vector defined by using the concepts of the WRO shown in Figure 1. Suppose we have a UDR containing 22 vacation packages, each annotated as shown in Table 1.

Table 1. Annotation of vacation packages based on ontology feature vector

ofv_1=*(InternationalHotel, FrenchMeal, Cinema, Flight)*	ofv_{12}=*(SeasideCottage, VegetarianMeal, Shopping)*
ofv_2=*(Pension, VegetarianMeal, ArtGallery, ShoppingCenter)*	ofv_{13}=*(Campsite, IndianMeal, Museum, RockConcert)*
ofv_3=*(CountryResort, MediterraneanMeal, Bus)*	ofv_{14}=*(RegularAcc, RegularMeal, Museum, Bazaar)*
ofv_4=*(CozyAcc, VegetarianMeal, Museum, Train)*	ofv_{15}=*(InternationalHotel, PictureGallery, Flight)*
ofv_5=*(InternationalHotel, ThaiMeal, IndianMeal, Concert, Bus)*	ofv_{16}=*(Pension, LightMeal, ArcheologicalSite, CarRental, Flight)*
ofv_6=*(SeasideCottage, LightMeal, ArcheologicalSite, Flight, ShoppingCenter)*	ofv_{17}=*(AlternativeAcc, LightMeal, RockConcert,Bus)*
ofv_7=*(RegularAcc, RegularMeal, Salon, Flight)*	ofv_{18}=*(CozyAcc, VegetarianMeal, Exhibition, ArcheologicalSite, Train)*
ofv_8=*(InternationalHotel, VegetarianMeal, Ship)*	ofv_{19}=*(CountryResort, VegetarianMeal, Concert, Bus)*
ofv_9=*(FarmHouse, MediterraneanMeal, CarRental)*	ofv_{20}=*(Campsite, MediterraneanMeal, ArcheologicalSite, Attraction, CarRental)*
ofv_{10}=*(RegularAcc, EthnicMeal, Museum)*	ofv_{21}=*(AlternativeAcc, LightMeal, Concert, Bus)*
ofv_{11}=*(RegularAcc, LightMeal, Cinema, Bazar)*	ofv_{22}=*(FarmHouse, LightMeal, RockConcert, Train)*

Consider the feature *VegetarianMeal* (a leaf in the taxonomy tree) which appears in each of the six *ofv*s of the set $\{ofv_2, ofv_4, ofv_8, ofv_{12}, ofv_{18}, ofv_{19}\}$ (see Table 1). The relative frequency of this feature over the whole resources (the 22 packages) is $w_f(VegetarianMeal) = 6/22 = 0.27$ (see Figure 1).

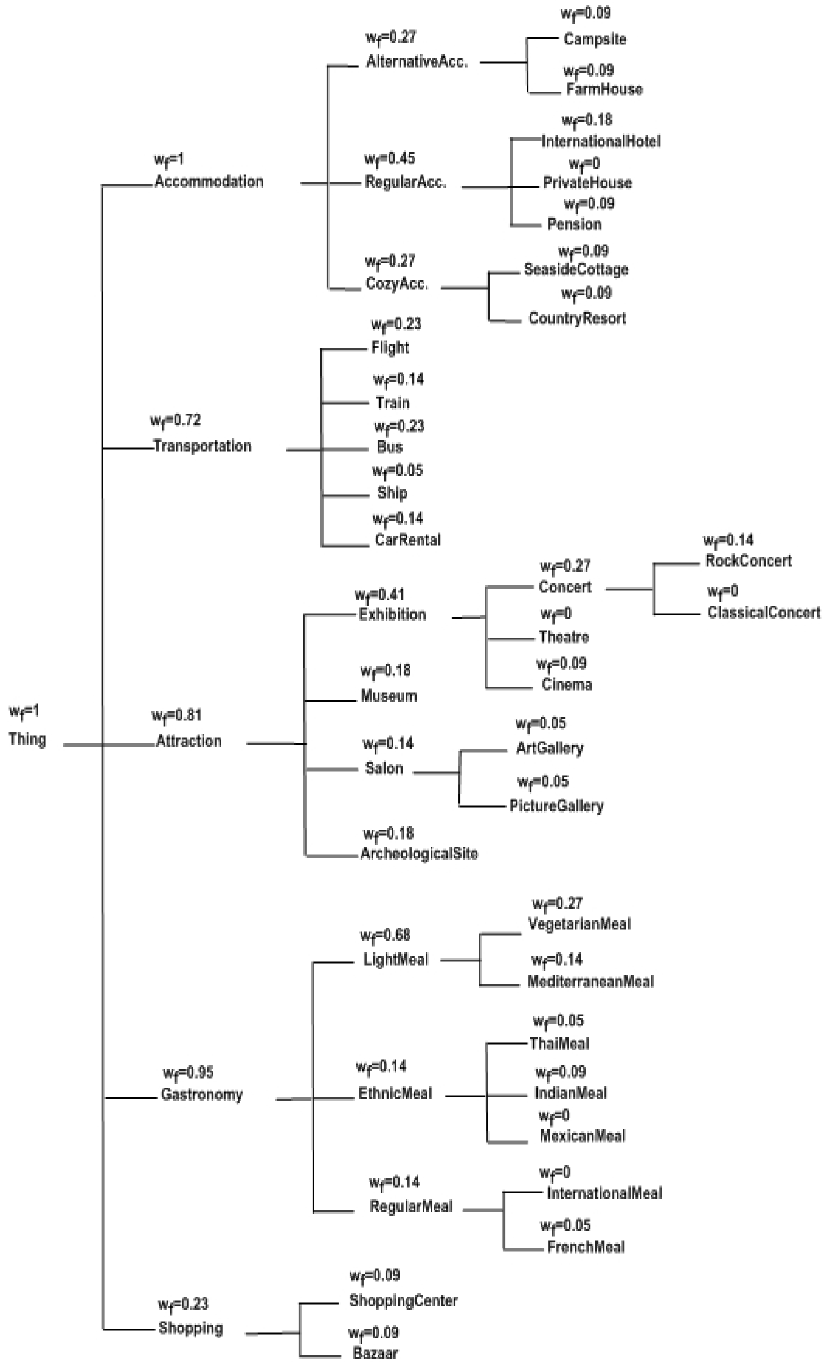


Fig. 1. WRO of our running example

In order to assign a weight to a more general concept, we have to consider not only the set of *ofv*s containing the concept but also the sets of *ofv*s containing its descendants in the ISA hierarchy. For instance, consider *LightMeal* which has only two children as descendants, namely *VegetarianMeal* and *MediterraneanMeal*. *MediterraneanMeal* appears in the *ofv*s of the set $\{ofv_3, ofv_9, ofv_{20}\}$. *LightMeal* is contained in the *ofv*s of the set $\{ofv_6, ofv_{11}, ofv_{16}, ofv_{17}, ofv_{21}, ofv_{22}\}$, therefore its weight is given by the cardinality of the union of the three sets above, divided by the total number of ofvs, i.e., $w_f(LightMeal)=15/22=0.68$. Note that, in the case of overlapping sets of *ofv*s, the union set in the above formula guarantees that w_f does not exceed one.

Let us consider a tourist who wants to visit a European capital and, in order to buy a vacation package, he/she expresses the following preferences:

"I would like travel by bus, sleep in a campsite, have ethnic food, and enjoy rock concerts".

It can be formulated according to the request feature vector notation as follows:

rv_1 = (*Campsite, EthnicMeal, RockConcert, Bus*).

The *SemSim* method allows the user to choose among the vacation packages offered by the tourism agency of our running example (shown in Table 1) the one that better satisfies his/her needs. In particular, it evaluates the similarity between feature vectors (*semsim*), which is based on the notion of similarity between concepts (features), referred to as *consim.*

Given a WRO, the *consim* method relies on the information content approach defined by Lin [11]. According to the standard argumentation of information theory, the information content of a concept c is defined as $-log\ w(c)$, where w is the weight associated with the concept c in the WRO. Therefore, as the weight of a concept increases the informativeness decreases hence, the more abstract a concept the lower its information content. Given two concepts c_i and c_j, their similarity, $consim(c_i,c_j)$, is defined as the maximum information content shared by the concepts divided by the sum of the information contents of the concepts. Note that we assume the ontology is a tree, therefore the least upper bound of c_i and c_j, $lub(c_i,c_j)$, is always defined and provides the maximum information content shared by the concepts in the taxonomy. Formally, we have:

$$consim(c_i, c_j) = 2 \times \frac{log\ w_f(lub(c_i, c_j))}{log\ w_f(c_i) + log\ w_f(c_j)}$$

For instance, consider the pair of concepts *VegetarianMeal* and *MediterraneanMeal* of the WRO shown in Figure 1. The *consim* measure between *VegetarianMeal* and *MediterraneanMeal* is defined as follows:

$$consim(VegetarianMeal,MediterraneanMeal) = 2\ log\ w_f\,(LightMeal)\ /\ (log\ w_f\,(VegetarianMeal) + log\ w_f\,(MediterraneanMeal)) = 0.24.$$

In fact, *LightMeal* is the *lub* between *VegetarianMeal* and *MediterraneanMeal*, and therefore provides the maximum information content shared by the comparing concepts.

As anticipated, the *SemSim* method allows us to compute the semantic similarity between the request vector *rv* and an *ofv*, *semsim(rv,ofv)*, by using the *consim* similarity above. We start by computing the *consim* for each pair of concepts of the Cartesian product of the two vectors. However, we are not interested in all possible pairs because in many cases the comparison is meaningless (e.g., contrasting *VegetarianMeal* with *Concert*). Hence, we restrict our analysis to the pairs that exhibit high affinity. In particular, we consider the *stable marriage problem* [14], which is the problem of finding a stable matching (i.e., there does not exist any better alternative pairing) between sets of elements. For instance, assuming *rv* and *ofv* represent a set of boys and a set of girls respectively, we analyze all possible sets of marriages, when polygamy is not allowed. Our solution, for the computation of the semantic similarity makes use of the Hungarian algorithm as an implementation for solving the *maximum weighted matching* problem in bipartite graphs [9] which runs in polynomial time. Essentially, the method aims to identify the set of pairs of concepts of the two vectors that maximizes the sum of the *consim* similarity values:

$$semsim(rv, ofv) = \frac{max(\sum consim(c_i, c_j))}{max(n, m)}$$

where: $i = 1..n$, $j = 1..m$, $n = |rv|$, $m = |ofv|$, $c_i \in rv$, and $c_j \in ofv$.

For instance, in the case of the request vector rv_1 defined above and the ofv_{21} shown in Table 1, the following set of pairs of concepts (enclosed in parenthesis) has the maximum *consim* sum, as shown below:

consim(*Campsite, AlternativeAcc.*) = 0.70
consim(*EthnicMeal, LightMeal*) = 0.04
consim(*RockConcert*, *Concert*) = 0.79
consim(*Bus*, *Bus*) = 1

Therefore: $semsim(rv_1, ofv_{21})$ = (0.70 + 0.04 + 0.79 + 1) / 4 = 0.63, where the sum of *consim* has been normalized according to the maximum cardinality of the contrasted vectors (in this case 4).

4 Semsim Experimental Analysis

In [5] we have presented an experiment where *SemSim* has been compared to some of the most representative similarity methods defined in the literature. In the experiment four request vectors have been defined, rv_i, i = 1,...,4, corresponding to the vacation preferences of four different tourists, which are recalled in Table 2. The correlation between each selected similarity method and human judgment (*HJ*) has been evaluated, where *HJ* values had been obtained by asking to a group of 21 people, randomly selected from our Institute staff, to indicate the similarity among each rv_i and the vacation packages ofv_i, j= 1,...,22, defined in Table 1. As already mentioned, in [5] our experiment has shown that the *SemSim* method based on the frequency approach has a higher correlation with *HJ* than the other approaches selected in the literature. Therefore, in this section, our analysis is focused on the *SemSim* method based on the frequency approach.

In Table 3, the *SemSim* values obtained according to our experiment are recalled. The *ofv*s are listed with the associated similarity values, starting from the highest up to the lowest values.

Table 2. Request Vectors

rv_1 = (Campsite, EthnicMeal, RockConcert, Bus)
rv_2 = (InternationalHotel, InternationalMeal, ArtGallery, Flight)
rv_3 = (Pension, MediterraneanMeal, Cinema, ShoppingCenter)
rv_4 = (CountryResort, LightMeal, ArcheologicalSite, Museum, Train)

Let us consider the first five *ofv*s listed in Table 3, i.e., the *ofv*s associated with the five highest *SemSim* values. For each rv_i $i = 1,...,4$, a table has been defined, namely Tables 4, 5, 6, and 7 respectively, where the *ofv*s with the five highest values have been indicated. In each table, in the first row the features of the addressed request vector are listed, and in the remaining rows, the features of the selected *ofv*s are recalled. The positioning of the features of the *ofvs* in the rows has been performed in order to show the set of pairs of features (one belonging to the request vector and the other one to the *ofv*) leading to the maximum weighted matching value, according to the stable marriage problem recalled above. In the rows, the features are shown with the related *consim* values (in square brackets) and a label (one among M, H, S, C and R) specifying the kind of relation existing between the matched features, as explained below:

- M stands for *exact match*, therefore the *consim* value is 1.0;
- H stands for *hierarchical* relation, i.e. one feature is a descendant/ancestor of the other; in parenthesis the number of arcs connecting the two features is also given (e.g., H(1) stands for a parent/child relation);
- S stands for *sibling*, i.e., children of the same father;
- C stands for *cousin*, i.e., children of siblings;
- R stands for *related*, for all the remaining pairs of features of the ontology.

Let us consider for instance Table 4, related to the request vector rv_1. According to the maximum weighted matching problem, the *ofv* with the highest *SemSim* value with rv_1 is ofv_{13}, which has four features. The pairs of features providing this value are (*Campsite*, *Campsite*), (*EthnicMeal*, *IndianMeal*), (*RockConcert*, *RockConcert*), and (*Bus*, *Museum*). Therefore, two features, namely *Campsite* and *RockConcert*, match exactly (M), *IndianMeal* is in the hierarchical relation H(1) with *EthnicMeal*, because the former is a child of the latter, and the other feature, *Museum*, is related to *Bus* according to the cousin relation (C). In square brackets we represent the *consim* values associated with these pairs, which in the case of the pair (*Bus*, *Museum*) is zero because these features do not share any information in the ontology except for *Thing*. In analyzing the four tables, as a general observation, note that the *ofv*s with higher *SemSim* values show a greater number of pairs of features with exact matches (M) with respect to the other *ofv*s. Immediately lower *SemSim* values correspond to a less number of exact matches M and a high number of parent/child relations H(1). Going down in the tables, the lowest *SemSim* values are associated with a higher number of related features (R), because R does not represent any specific relation between the features in the ontology. For instance, in the case of the request vector rv_2, in the first row of Table 5 we have the ofv_{15} showing two exact matches and one sibling relation, and in the second row we find the ofv_7, for which one single match and three parent/child relations are present.

Table 3. Ranking results for request vectors

rv_1		rv_2		rv_3		rv_4	
Ranked Resources	*Value*	*Ranked Resources*	*Value*	*Ranked Resources*	*Value*	*Ranked Resources*	*Value*
ofv_{13}	0.73	ofv_{15}	0.66	ofv_{11}	0.61	ofv_{18}	0.66
ofv_{17}	0.69	ofv_7	0.60	ofv_2	0.58	ofv_4	0.63
ofv_{21}	0.63	ofv_1	0.52	ofv_1	0.35	ofv_6	0.55
ofv_5	0.54	ofv_2	0.35	ofv_{14}	0.31	ofv_{16}	0.44
ofv_{19}	0.45	ofv_{16}	0.29	ofv_6, ofv_{16}	0.28	ofv_{22}	0.42
ofv_{22}	0.44	ofv_8	0.28	ofv_3, ofv_9, ofv_{12}	0.25	ofv_{19}	0.35
ofv_{10}, ofv_{20}	0.28	ofv_5	0.26	ofv_{20}	0.23	ofv_{20}	0.34
ofv_3	0.26	ofv_6	0.22	ofv_{21}	0.20	ofv_3	0.30
ofv_9	0.19	ofv_{10}, ofv_{11}, ofv_{14}	0.18	ofv_5, ofv_{17},	0.18	ofv_{17}, ofv_{21}	0.26
ofv_{18}	0.17	ofv_{19}, ofv_{21}	0.08	ofv_{19}, ofv_{22}		ofv_{13}	0.23
ofv_1	0.16	ofv_4, ofv_{17}, ofv_{22}	0.07	ofv_{18}	0.16	ofv_{11}	0.22
ofv_{11}	0.11	ofv_{18}, ofv_{20}	0.06	ofv_7, ofv_8, ofv_{10}	0.15	ofv_{10}, ofv_{14}	0.21
ofv_4, ofv_7	0.08	ofv_3, ofv_9	0.05	ofv_{13}, ofv_{15}	0.11	ofv_{12}	0.20
ofv_6, ofv_{15}, ofv_{16}	0.07	ofv_{13}	0.02	ofv_4	0.08	ofv_8	0.12
ofv_8	0.04	ofv_{12}	0.00			ofv_2	0.11
ofv_2, ofv_{14}	0.03					ofv_9	0.10
ofv_{16}	0.01					ofv_5, ofv_7	0.07
						ofv_1	0.06
						ofv_{15}	0.05

However, this does not hold in general. In fact in the same table, ofv_1 in the third position has one more exact match with respect to ofv_7. The reason is that, in the case of ofv_1, two exact matches are combined with a cousin relation which has a very low *consim* value. We have a similar situation in Table 6, for the ofv_{11} and ofv_2 in the case of the request vector rv_3. In fact, the combination of one exact match (*Cinema*) with two parent/child relations (*Pension*, *RegularAcc)* and (*MediterraneanMeal*, *LightMeal*) leads to a higher *SemSim* value for the ofv_{11} with respect to the ofv_2 for which two exact matches (*Pension*) and (*ShoppingCenter*) are combined with a sibling relation (*MediterraneanMeal*, *VegetarianMeal*) and a cousin relation (*Cinema*, *ArtGallery*), the last one with a very low value.

Table 4. Ranking results for rv_1

rv_1	*Campsite*	*EthnicMeal*	*RockConcert*	*Bus*		**Semsim**
ofv_{13}	Campsite M [1.0]	IndianMeal H(1) [0.91]	RockConcert M [1.0]	Museum C [0]		**0.73**
ofv_{17}	AlternativeAcc H(1) [0.70]	LightMeal S [0.04]	RockConcert M [1.0]	Bus M [1.0]		**0.69**
ofv_{21}	AlternativeAcc H(1) [0.70]	LightMeal S [0.04]	Concert H(1) [0.79]	Bus M [1.0]		**0.63**
ofv_5	InternationalHotel C [0]	IndianMeal H(1) [0.91]	Concert H(1) [0.79]	Bus M [1.0]	ThaiMeal [0]	**0.54**
ofv_{19}	CountryResort C [0]	VegetarianMeal R [0.03]	Concert H(1) [0.79]	Bus M [1.0]		**0.45**

Consider now the *ofv*$_5$ in Table 4. It contains one more feature than the request vector *rv*$_1$ and, according to the stable marriage problem, it will necessarily have a dangling feature in the table. This feature is *ThaiMeal*, even though in the request vector there is *EthnicMeal* which is its parent. This is due to the higher *consim* similarity value between *IndianMeal* and *EthnicMeal* (0.91) than *ThaiMeal* and *EthnicMeal* (0.78). For this reason, since *ThaiMeal* does not contribute in any way to *SemSim* value, the value in square brackets has been set to zero.

Table 5. Ranking results for *rv*$_2$

***rv*$_2$**	***InternationalHotel***	***InternationalMeal***	***ArtGallery***	***Flight***		**Semsim**
***ofv*$_{15}$**	InternationalHotel M [1.0]		PictureGallery S [0.64]	Flight M [1.0]		**0.66**
***ofv*$_7$**	RegularAcc H(1) [0.63]	RegularMeal H(1) [0]	Salon H(1) [0.78]	Flight M [1.0]		**0.60**
***ofv*$_1$**	InternationalHotel M [1.0]	FrenchMeal S [0]	Cinema C [0.07]	Flight M [1.0]		**0.52**
***ofv*$_2$**	Pension S [0.38]	VegetarianMeal C [0]	ArtGallery M [1.0]	Shopping Center R [0]		**0.35**
***ofv*$_{16}$**	Pension S [0.38]	LightMeal R [0]	ArcheologicalSite R [0.08]	Flight M [1.0]	CarRental [0]	**0.29**

Note that in the case of a feature associated with a weight equal to zero, such as for instance *InternationalMeal* (see the ontology in Figure 1), we assume that *consim* between this feature and any other feature is equal to zero as well. This is the reason why in Table 5, in the case of the *ofv*$_7$, the pair *RegularMeal, InternationalMeal* has a *consim* value equal to zero, even though they are in a parent/child relation (and the same results are obtained for the other features in the *InternationalMeal* column with different kinds of relations).

Table 6. Ranking results for *rv*$_3$

***rv*$_3$**	***Pension***	***MediterraneanMeal***	***Cinema***	***ShoppingCenter***		**Semsim**
***ofv*$_{11}$**	RegularAcc H(1) [0.49]	LightMeal H(1) [0.32]	Cinema M [1.0]	Bazaar S [0.62]		**0.61**
***ofv*$_2$**	*Pension* M [1.0]	VegetarianMeal S [0.23]	ArtGallery C [0.07]	ShoppingCenter M [1.0]		**0.58**
***ofv*$_1$**	InternationalHotel S [0.39]	FrenchMeal C [0.02]	Cinema M [1.0]	Flight R [0]		**0.35**
***ofv*$_{14}$**	RegularAcc H(1) [0.49]	RegularMeal R [0.02]	Museum R [0.10]	Bazaar S [0.62]		**0.31**
***ofv*$_6$**	SeasideCottage C [0]	LightMeal H(1) [0.32]	ArcheologicalSite R [0.10]	ShoppingCenter M [1.0]	Flight [0]	**0.28**
***ofv*$_{16}$**	Pension M [1.0]	LightMeal H(1) [0.32]	ArcheologicalSite R [0.10]	CarRental R [0]	Flight [0]	**0.28**

Table 7. Ranking results for *rv*$_4$

***rv*$_4$**	***CountryResort***	***LightMeal***	***Museum***	***ArcheologicalSite***	***Train***	**Semsim**
***ofv*$_{18}$**	CozyAcc H(1) [0.70]	VegetarianMeal H(1) [0.46]	Exibition S [0.15]	ArcheologicalSite M [1.0]	Train M [1.0]	**0.66**
***ofv*$_4$**	CozyAcc H(1) [0.70]	VegetarianMeal H(1) [0.46]	Museum M [1.0]		Train M [1.0]	**0.63**
***ofv*$_6$**	SeasideCottage S [0.54]	LightMeal M [1.0]	ShoppingCenter R [0]	ArcheologicalSite M [1.0]	Flight S [0.18]	**0.55**
***ofv*$_{16}$**	Pension C [0]	LightMeal M [1.0]	CarRental C [0]	ArcheologicalSite M [1.0]	Flight S [0.18]	**0.44**
***ofv*$_{22}$**	FarmHouse C [0]	LightMeal M [1.0]		RockConcert R [0.11]	Train M [1.0]	**0.42**

Note that in some cases the *ofv*s may have high *SemSim* values even though in the presence of dangling features of the request vector. This is the case, for instance, of the ofv_{15} in Table 5, which is associated with the highest *SemSim* value in the list (0.66). In fact, it presents two exact matches (*InternationalHotel*) and (*Flight*), one sibling relation (*ArtGallery*, *PicureGallery*), and the *InternationalMeal* feature of the request vector rv_2 that is dangling.

5 Visualization Methods for Results

In order to provide a better visualization of the results of the *SemSim* method, we show them in graphical ways. In particular, we adopt radial and radar representations. The former gives a general idea of the results, while the latter focuses on a particular *ofv*. In the radial representation, given a request vector (rv) we place the *SemSim* values for the different *ofv*s on concentric circumferences where the radius of each circumference corresponds to the *SemSim* value of the related *ofv*. In Figure 2, an example related to the rv_4 is given, where the first five *ofv*s in the ranking have been reported according to Table 7.

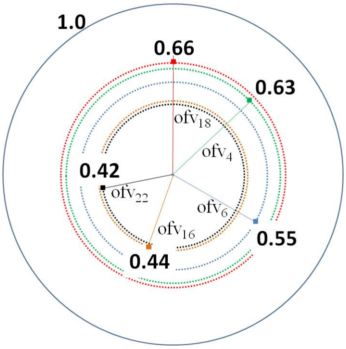


Fig. 2. Radial representation for rv_4

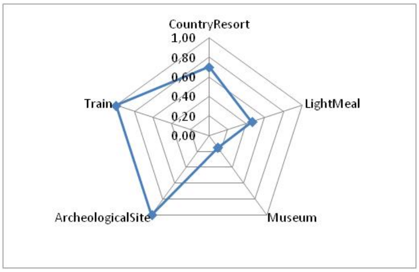


Fig. 3. Radar representation (ofv_{18}, rv_4)

In the radar representation, given a *rv* and an *ofv*, we show all the features of the *rv*, and for each of these features, the *consim* value obtained from the pairs formed with the concepts of the *ofv*, as shown in Table 7. For instance, rv_4 is composed by five features: *CountryResort*, *LightMeal*, *Museum*, *ArcheologicalSite*, and *Train*. Therefore, rv_4 is represented as a pentagon, where each vertex is labeled with one of its features. As illustrated in Figure 3, the radar representation shows how the rv_4 is approximated by the ofv_{18}.

6 Conclusions

In this paper, we have addressed *SemSim*, a method for the semantic search and retrieval of resources that populate a search space. *SemSim*, which has been already experimented in the tourism sector, has been further investigated. In particular, a qualitative analysis has been given by providing the user with an explanation about the mechanism adopted in deriving the ranked list of objects. Furthermore, graphical representations about the results have also been given.

As a future work, we plan to study ontology comprising different types of ontological relations.

References

1. Batzios, A., Mitkas, P.A.: WebOWL: A Semantic Web search engine development experiment. Expert Systems with Applications (ESWA 2012) 39(5), 5052–5060 (2012)
2. Castano, S., De Antonellis, V., Fugini, M.G., Pernici, B.: Conceptual Schema Analysis: Techniques and Applications. ACM Transactions on Databases Systems 23(3), 286–333 (1998)
3. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer (2007)
4. Fang, W.-D., Zhang, L., Wang, Y.-X., Dong, S.-B.: Towards a Semantic Search Engine Based on Ontologies. In: Proceedings of 4th International Conference on Machine Learning, Guangzhou (2005)
5. Formica, A., Missikoff, M., Pourabbas, E., Taglino, F.: Semantic search for matching user requests with profiled enterprises. Computers in Industry 64, 191–202 (2013)
6. Francis, W.N., Kucera, H.: Brown Corpus Manual. Providence, Rhode Island. Department of Linguistics, Brown University (1979)
7. Gruber, T.R.: A translation approach to portable ontologies. Knowledge Acquisition 5(2), 199–220 (1993)
8. Kim, J.W., Candan, K.S.: CP/CV: Concept Similarity Mining without Frequency Information from Domain Describing Taxonomies. In: Proceedings of Conference on Information and Knowledge Management, CIKM 2006 (2006)
9. Kuhn, H.W.: Variants of the Hungarian method for assignment problems. Naval Research Logistics Quarterly 3, 253–258 (1956)
10. Lee, M., Kim, W., Park, S.: Searching and ranking method of relevant resources by user intention on the Semantic Web. Expert Systems with Applications (ESWA 2012) 39(4), 4111–4121 (2012)
11. Lin, D.: An Information-Theoretic Definition of Similarity. In: Shavlik, J.W. (ed.) Proceedings of 15th the International Conference on Machine Learning, pp. 296–304. Morgan Kaufmann, Madison (1998)
12. Maarek, Y.S., Berry, D.M., Kaiser, G.E.: An Information Retrieval Approach For Automatically Constructing Software Libraries. IEEE Transactions on Software Engineering 17(8), 800–813 (1991)
13. Maguitman, A.G., Menczer, F., Roinestad, H., Vespignani, A.: Algorithmic Detection of Semantic Similarity. In: Proceedings of World Wide Web Conference (WWW 2005) Conference, Chiba, Japan (May 2005)
14. Mairson, H.: The Stable Marriage Problem. The Brandeis Review 12(1) (1992)
15. Melo, D., Pimenta Rodrigues, I., Beires Nogueira, V.: Work Out the Semantic Web Search: The Cooperative Way. In: Advances in Artificial Intelligence (ADVAI 2012) (2012)
16. Miller, G.A.: Wordnet: A Lexical database for English. Communication of the ACM 38(11), 39–41 (1995)

An Application of Philosophy in Software Modelling and Future Information Systems Development

Brian Henderson-Sellers[1], Cesar Gonzalez-Perez[2], and Greg Walkerden[3]

[1] Faculty of Engineering and Information Technology,
University of Technology, Sydney, Broadway, NSW, Australia
[2] Institute of Heritage Sciences (Incipit), Spanish National Research Council (CSIC)
Santiago de Compostela, Spain
[3] Dept Environment & Geography, Macquarie University, North Ryde, NSW, Australia
brian.henderson-sellers@uts.edu.au,
cesar.gonzalez-perez@incipit.csic.es, greg.walkerden@mq.edu.au

Abstract. The influence of mainstream philosophy on conceptual modelling and on modelling language development has historically been arcane or, at best, not recognized, whilst modellers might in fact implicitly espouse one particular philosophical tenet. This paper describes and discusses philosophical stances applied to conceptual modeling in order to make such influences explicit so that we, as conceptual modellers, can take the next step.

Keywords: concepts, modelling, philosophy, ontology engineering, conceptual modeling.

1 Introduction

The influence of mainstream philosophy on modelling language development, an important element in the information systems development context, has historically been minimalistic or, if even a little influential, has been a hidden source. In other words, whilst many modellers might in fact be behaving as if they held one particular philosophical tenet, or perhaps several inconsistent philosophical stances, they are often unaware of the philosophical assumptions that are thus implicit in the way they construct their modelling theories and hence unaware of consequences that may be disruptive or unhelpful in the way they model.

This paper delves back into philosophical writings to make a variety of such influences explicit in order that conceptual modellers and ontology engineers[1] can take the next step in formalizing their knowledge base. The aim of the paper is, therefore, not to espouse any particular philosophical thinking as ‘correct’ for conceptual modellers and ontology engineers, but to highlight the fact that every information systems researcher and developer has tacit assumptions they are often unable to articulate and to

[1] Although these belong to different communities with different histories, the relationship of these two traditions to the explication of philosophical assumptions is fundamentally similar, such that no differentiation is drawn between them in this paper.

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 329–340, 2013.

propose that, by taking an explicit and conscious stance, modellers will be able to make better decisions about modelling. For example, an in-depth understanding of stereotypes [1] helps a modeller to avoid the common mistake of equating them with supertypes [2].

Adopting one particular philosophical stance may entail a number of assumptions regarding the relationships between elements in a conceptual model and whether certain concepts are included or excluded. For instance, whether types exist in the real world; whether properties exist or, if not, whether they can be pragmatically useful for conceptual modelling. Opting for one stance might preclude the inclusion of an element that essentially belongs to, and is only valid in, a different philosophical view. Alternatively, one could take a more pragmatic philosophical stance, and embrace a more philosophically eclectic approach, looking for benefits for conceptual modelling not previously available - although exploration of such possibilities is a topic for future research.

With regard to concepts, for example, a Lockean [3] model asserts that we think with a mix of "simple ideas" and "complex ideas". For Locke, a "simple idea" is an idea that comes straight from experience - we have sensory experience of the world around us, and we have direct experience of an 'interior realm'. A "complex idea", by contrast, is formed by combining simple ideas, by finding relationships between them, and/or by abstracting away from them [Bk2, Ch12, S1]. All knowledge derives ultimately from experience [Bk2, Ch1, S2] -footnote1-. For Locke, the sounds of words are “marks for ideas within [one’s] own mind” [Bk3, Ch1, S1].

Each of Locke's key points has been a focus for extensive philosophical debate. For instance it has been argued: 1. There must be a knower for there to be knowledge, and the nature of the knower must influence what knowledge is, so all knowledge cannot be derived *wholly* from experience. 2. All ideas are more like "complex ideas" in Locke's terms: the simplest concepts embody rich experience, and function metaphorically. 3. It does not make sense to talk of ideas as if they existed independently of words or other symbols.

The debates about these points provide alternative points of departure for conceptual modellers. Contemporary interpretations of concepts, e.g. as discussed by [4], provide still more. We advocate sensitivity to these opportunities: a sensitivity to philosophical assumptions can bring clarity to ambiguities and contradictions in methods of software modelling and information systems development, and lead to innovations.

Although we originally sought to identify the ‘best’ philosophical underpinnings for software development and conceptual modelling, a study of the philosophical literature reveals not only differences between philosophy of the mind, philosophy of language use and philosophy of psychology – all relevant to our study – but also that no one of these philosophies can be considered ‘correct’. In other words, we can only describe and discuss these philosophical stances in the context of conceptual modelling; we can only recommend that modellers and modeling language designers should consider either (1) taking a systematic philosophical approach adopting a consistent set of assumptions, consistent from one particular philosophical viewpoint; or (2) embracing a more pragmatic approach, combining philosophical insights that are helpful for the task at hand and focusing on becoming aware of previously unrecognized constraints and possible missed opportunities.

In Section 2, we discuss some of the (historical) involvement of philosophical thinking with conceptual modelling including how varying philosophical assumptions may have unnoticed repercussions. In Section 3, we discuss some of the ideas found in the ontology engineering literature, noting the minimal overlap with philosophy whilst Section 4 presents our conclusions.

2 Philosophical Impacts on Modelling

As noted above, the impact of choosing one particular philosophical stance may exclude (or demand) certain concepts, ideas and representations. For instance, [5, Ch 5] argues for a 'logical paradigmatic' philosophical viewpoint, in which it is forbidden (incorrect) to include representations of properties within any model thus constructed. That this is counter-intuitive to accepted conceptual modelling (e.g. using UML) provides a tension between contemporary philosophy and conceptual modelling that requires future resolution (see Section 2.2).

Lakoff [6] argues from a cognitive linguistic viewpoint rather than a strictly philosophical basis that human reason is embodied symbol manipulation, noting that, through the classical theory of categories, symbols acquire meaning by virtue of their correspondence to categories in the real world. Since symbols are important for conceptual modelling languages, this notion of how symbols relate to, refer to or represent 'entities in the real world' is one of the crucial elements of the philosophical impacts on conceptual modelling.

2.1 Concepts

One area of major contention is in the notion of 'concept' itself (and 'conceptualization' and 'conceptual' – as in 'conceptual modelling'). The Stanford Encyclopedia of Philosophy [7] identifies, *inter alia*[2], three main options for discussing concepts: concepts as mental expressions, dating back to Locke [9] and Hume [10]; concepts as abilities [11]; and concepts as Fregean senses e.g. [12]. It states:

> *"Concepts are the constituents of thoughts. Consequently, they are crucial to such psychological processes as categorization, inference, memory, learning, and decision-making. This much is relatively uncontroversial. But the nature of concepts—the kind of things concepts are—and the constraints that govern a theory of concepts have been the subject of much debate."*

Smith [4], whilst decrying the over- or mis-use of 'concept', states: "In many contexts, of course, ontologists still deal with concepts, correctly, as analogous to, though more abstract than, the linguistic expressions with which they are associated." His concern centres as much around the misapplication of the notion of concept rather than the notion itself, observing that many researchers deal only with concepts rather than the entities in reality to which they correspond.

---

[2] Other possibilities are discussed in [8].

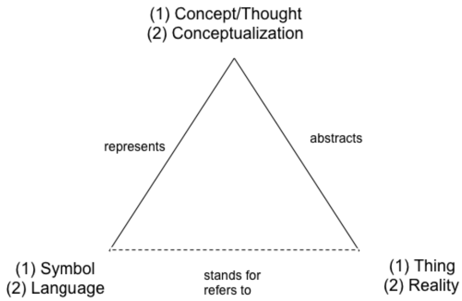


Fig. 1. Ogden and Richards' 'meaning triangle' (a.k.a. Ullmann's triangle) for (1) an individual Thing and (2) reality (or a specific reality domain such as banking or telecomms)

Further debate has emphasized two alternatives:

1. The concept strategy, as discussed above. If we wish to relate this to Fig. 1, we might argue that the three vertices of the triangle are utilized in such a way that 'concept' (the topmost vertex) effectively dominates and relegates the notion of sign or symbol to be a 'second class citizen'.
2. The sign strategy, which, whilst acknowledging mental constructs, notes that these are individualistic and that we should be most concerned about communication – via signs and symbols. In terms of Fig. 1, only the bottom part of the figure would be relevant. To date the former has been more popular in ontology engineering and the latter in UML-style software modelling.

Option 2 is favoured by Smith [4], who argues that the misuse of 'concept' has resulted from its use in the knowledge representation community and in linguistics where concepts and entities tend to become confounded. In ontological information systems, a prime reference is that of Gruber [13] who famously defined ontology as "a specification of a conceptualization", thus freezing in the notions of concept and conceptualization into the very fabric of ontology research – incorrectly according to much philosophical thinking. It should be stressed that the semantics of 'ontology' in these three cases (Smith, Bunge and Gruber) are by no means identical – another example of the challenge to conceptual modellers of employing a consistent philosophical and ontological framework.

In ontology engineering, it is too often the case, he argues [13], that the ontology being created is of social constructs, which have some verity in being labelled 'concepts', being largely human-created and intangible (see also [14]); whereas in the broader picture, it is vital that elements in the ontology have a direct correspondence to entities in the real world [15]. Option 2 also has the advantage of parsimony – a good trait in science[3]. Smith [4] thus strongly advocates replacing the emphasis on concepts and conceptualization by universals and particulars such that particulars refer to instances that exist in reality and universals signify what the corresponding instances have in common. In object-oriented conceptual modelling, the typical

[3] Usually referred to as Ockham's razor, named after William of Ockham, c 1287-1347.

approach is to relate particulars to objects – although in fact they could equally refer to classes of objects – and universals to classes (Fig. 2). This, again, offers support, this time from philosophy for the shift of emphasis in metamodelling, as suggested in [16] to a framework focussing on language use as determining the modelling architecture rather than a strict metamodelling hierarchy as advocated by the OMG (see also discussion in Section 3.2).

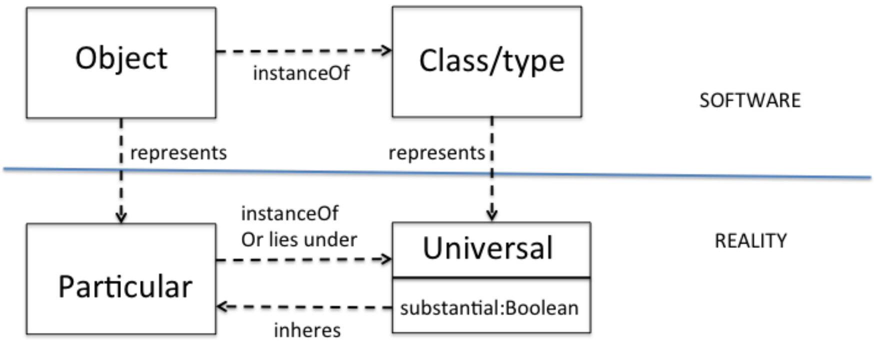


Fig. 2. A depiction of elements in the software domain and how they represent elements of reality

This provides a way to differentiate good ontologies from bad ontologies, differentiating them on the basis of whether the general terms in them do or do not relate to corresponding universals (and hence instances) in reality. Smith [4] also argues for the universal-instance model to be taken as axiomatic for ontologies and then investigates the meaning of subtyping (necessarily between universals) and mereology (linking particulars).

2.2 Properties

As well as concepts, an additional philosophical issue is that of the nature, utility and existence of properties[4], common in modelling languages like UML (see, e.g., discussion in [17]), in ontologies (e.g. the property of Bunge [18] and Wand [19]; see also [20]) and in cognitive linguistics (e.g. [4]). In philosophy, it is argued [21] that properties are crucial to analytic philosophy, although other philosophers disagree [22], arguing that properties are eschewed in contemporary philosophy.

Properties were originally identified with 'substance' (e.g. [3, chapter 4]), an idea that was strongly attacked by both Locke [9] and Hume [10] yet strongly supported in contemporary foundational ontologies, such as the UFO [24] and the four-category ontology of Lowe [25].

For some philosophers, this resulted in a shift from a substance paradigm to an extension paradigm e.g. [26], along with which came a shift (in philosophical thinking) from properties to logical classes and tuples e.g. [5, page 94]. However, this

[4] Properties are sometimes called qualities or characteristics e.g. [23, p105]. In the software engineering context, properties are often called attributes.

necessary realignment of properties as extensions brings its own challenges and contradictions. As an example, consider the red property of an object that is turned into a member of a class of RedThings and compare this with the representation of a ‘redness’ quality as a trope [27]. Consequently, we do not advocate the adoption of such philosophical traits coming from the extension paradigm into conceptual modelling; rather, we retain the notion of property since this is all-pervasive in the software and conceptual modelling and metamodelling literature to date. This is consistent with the use of the Bunge ontology by Wand and Weber [28-30]: the BWW model applied to information systems development.

3 Some Contributions of Ontological Engineering and Language Use

3.1 Ontological Engineering – A Philosophical Perspective

There is a long history (see, e.g. [31, 32]), identified in the ontological engineering literature (see, e.g., [33]), dating back to Aristotle and summarized more recently by Ogden and Richards [34], that suggests that there is a mediating role of the human mind in relating symbols to things (in reality). This leads directly to a representation such as that in Fig. 1 (with label ‘1’). Based on [35] but known as the Ullmann triangle [36], it illustrates the well-known ‘meaning triangle’ linking an object or thing in the real world to a concept or thought and then to a symbol (thus bearing some similarity with Locke’s approach outlined above); in other words, that the linking of model to reality is not direct. For a given domain (part or all of reality), the set of all the individual concepts abstracted from that domain is called the *conceptualization* (Fig. 1 (label ‘2’)) i.e. the combination of the concepts and their relationships [33]. This is the *mental model* e.g. [37, 38] – called here the *cognitive model*– see also [39] and could be said to imply and embody the ontological commitment (e.g. [40]). However, in this work, the conceptualization is considered as an *individual* mental commitment that must be shared (at some symbolic level) in order to be useful to the community; whereas in other more commonly accepted philosophies, individual mental commitments, whatever their appellation, are deemed unhelpful unless they are confirmed and agreed upon by the community who wish to share and build on these ideas i.e. conceptualization relates to the community accepted description not to the individual’s.

However, whilst often used in ontological discussions (e.g. [33, 41]), we must point out that this work (of Peirce, Ogden, Nash and Ullman) is not appreciated in any of the philosophical literature and emerges more from formalising everyday usage than disciplined philosophical reflection. Nevertheless, from our pragmatic point of view, the ideas, which conflate types and instance, are nevertheless worth further investigation because the simple model of Fig. 1 may provide a starting point for a detailed mathematical description of the modelling and metamodelling domains [15].

Fig. 3 suggests a much more detailed description to tie together cognition, reality and models. The first triangle deals with the physical world, which is made of things.

Each thing in the world (such as the chair I am sitting on as I type this) may be perceived by us through an individual concept and I can utter words or use any other symbols (graphical, aural or otherwise) to refer to it. For example, the word 'chair' in the utterance "this chair is getting older" pronounced by me while I keep the mental picture of my chair active is a symbol usage. Moving to the conceptual world, things can be arranged into categories or thing types; for example, all chairs conform to (or 'fall under') a *chair* thing type, and we often use that category for abstraction purposes. Our mental picture of that category is what we call the corresponding type concept; different people may have different mental pictures of the same category (or thing type), although most of us would probably share a common thing type for *chair*. In addition, categories are also things and thus can be seen as individual concepts and described by symbol usages. The third component of the conceptual world is given by the symbols that we may use to depict the categories in it through the corresponding type concepts; for example, the word 'chair' in English, which does not represent any particular chair but the very concept of chair, is a symbol. Moving further on to the modelling world, we can conceive a model as an aggregate of symbol usages; for example, an uttered sentence (which is a model) is an aggregate of particular uttered words, and a UML model is an aggregate of particular usages of UML symbols. A language, similarly, is an aggregate of symbols and usage rules (e.g. grammar, semantics and pragmatics); English focusses on morphemes whereas UML is a collection of metamodel elements. And, finally, a conceptualization is an aggregate of concepts, i.e. a particular worldview on the physical world.

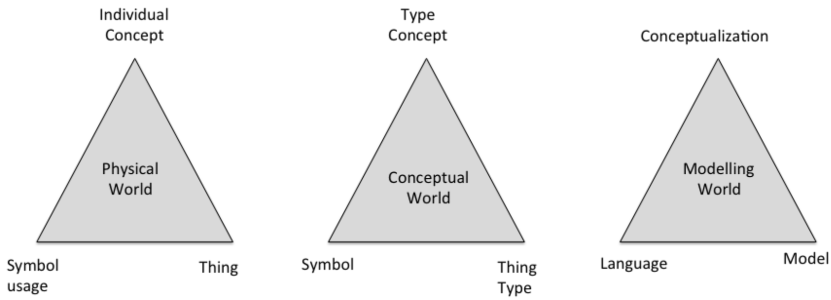


Fig. 3. Taking Fig. 1 as a base, the three triangles here explicitly differentiate the elements in the physical, conceptual and modelling worlds

The three-triangle approach to the description of meaning (Fig. 3) is, in our view, superior to the single triangle of Ogden and Richards, since it clearly differentiates between types and instances. In Fig. 1, instances and their types are conflated into a single element, which is confusing and avoids formal reasoning. For example: the English word 'chair', as found in a dictionary, and the uttered word 'chair' as part of a sentence that refers to a particular chair are definitely not the same kind of thing; the former is a symbol and the latter is a symbol usage; the former is part of a language and the latter is part of a model. In Fig. 1, this distinction goes unnoticed.

3.2 Language Use

The study of meaning in a natural language – so-called 'language use' – has a modern history of several decades (e.g. [23, 42]) and itself embodies a philosophical underpinning dating back to at least Frege [43]. The application of these ideas to conceptual modelling and information systems development is discussed in [16], where it is argued that the current multilevel architectures, such as that of the Object Management Group (OMG) in which four levels are linked by instanceOf relationships, should be replaced by an architecture based on language use in which elements in any model are not instances of classes in a metamodel but are in fact instances and/or types that conform to a modelling language definition (Fig. 4). In other words, the idea of a metamodelling stack in which the metamodel is the dominant feature (as in today's UML: [44]) is replaced by one in which the modelling language itself is the predominant feature [16]. Of course, this language may be defined by a metamodel but that definitional link is only one of several options.

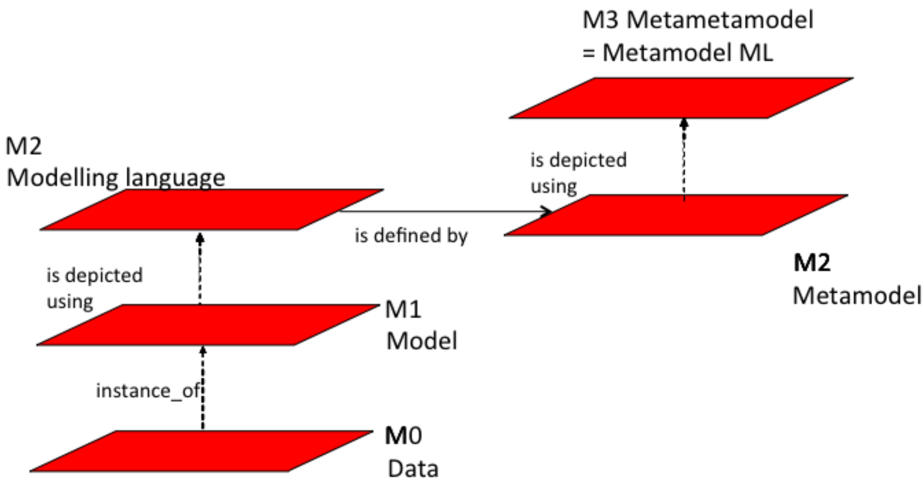


Fig. 4. Revision of the OMG multi-level architecture place the modelling language and not the metamodel at the 'M2' level in the multilevel stack (now restricted to three levels with a linkage to a second multilevel stack – here shown as having only two levels) (after [16])

Furthermore, this architecture fully supports models in the real world. This gives a correspondence relationship between particulars (instances) in the language use component of the model to real world things (Figs. 3 and 5). It is important to note that (i) this correspondence is generally ignored in conceptual modelling approaches in use today, especially those based on UML, and (ii) it introduces the need to recognize that there may be a need for a better framework. For example, in the former case, it is well-known that modellers have difficulty in identifying whether actors in a use case diagram are internal or external to the software system e.g. [45]. In the latter case, it is possible that the clumsiness of such models could be ameliorated by the use of such a theoretical framework that already exists, but outside of conceptual modelling/software engineering – for example, a philosophical underpinning between an entity (e.g. a particular or a universal) and a referent (e.g. [23, 43].

These ideas, of cognitive models (in an individual's brain) leading to a (shared) concept is well accepted in the modelling community and, indeed, in cognitive linguistics (e.g. [6, pp. 8 & 178]) and in language use/speech act theory, although Searle [23] notes that Frege's original use of the word concept confused two different definitions and usages resulting from "Frege moving in two philosophical directions which are at bottom inconsistent" [23, p98]: (a) insisting that predicates have a referent and (b) accounting for a functional distinction between referring expressions and predicate expressions. Searle argues for a solution in which the notion that predicate expressions refer is abandoned i.e. they ascribe properties they do not refer to them. Here, Searle seems to equate property with concept. Finally, we note that in the modelling area, there is a whole branch known as 'conceptual modelling' (e.g. [46, 47]), with its own eponymous conference. However, references to the Ogden and Nash/Ullmann triangle discussed above are all but absent in the philosophical research literature.

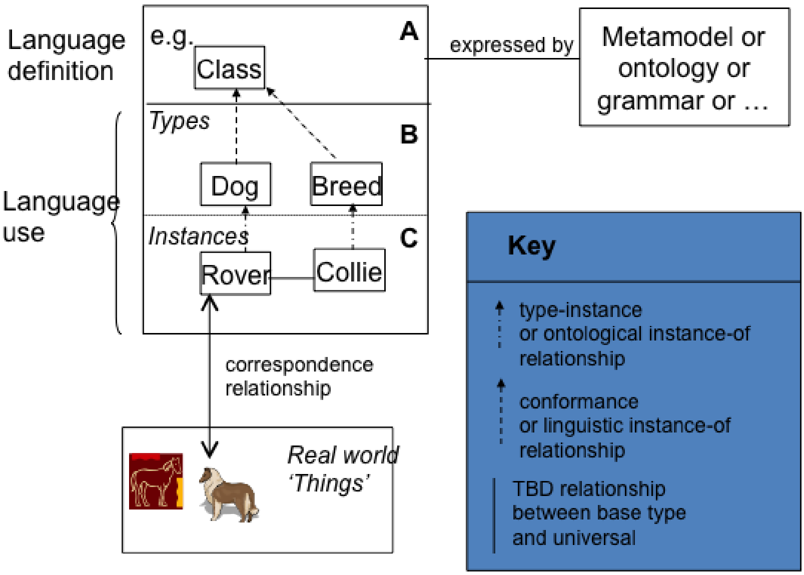


Fig. 5. Complete multilevel framework based on language use – to replace the strict metamodelling architecture of the Object Management Group when modelling in information systems development and software engineering (after [15])

3.3 Ontic vs. Epistemic Models

Conceptual models are supposed to represent things, but it is rarely clear whether the things represented by a conceptual model are the things-as-they-are or the things-as-we-know-them. A model of the things-as-they-are, also called an ontic model, aims to represent reality while being independent of the observer. A model of the things-as-we-know-them, on the other hand, aims to capture our knowledge about the reality rather than the aseptic reality itself. All but the most radical solipsists would agree that different degrees of subjectivity can be captured in a model; however, conventional conceptual modelling languages such as UML lack the necessary mechanisms to capture these nuances, and models created with them, in consequence, are oblivious to them. For example, we would certainly agree that "every person has an age", so that the property Age of class Person should not be nullable in a UML class model. However, a modeller might choose to make it nullable if there are chances that Person objects will be created without a known age. Note, however, that the first statement ("every person has an age, so that the property Age of class Person should not be nullable") is ontic in nature, whereas the second ("a modeller might choose to make it nullable if there are chances that Person objects will be created without a known age") is epistemic. The ConML modelling language [48, 49], for example, has separate *null and unknown semantics* in order to describe information that does not exist ("null", which is ontic) and information that does exist but which we ignore ("unknown", which is epistemic). Without this explicit difference, a model is confusing, since there is no way to know whether it describes an objectivized reality or a particular epistemic approach to it.

4 Summary and Conclusions

The role of conceptual modelling in information systems development has continued to increase over the last several decades. However, there has been little explicit awareness of any influences of mainstream philosophical thinking. This neglect could not only impact the cohesiveness of conceptual modelling paradigms (such as ER, OO) but also obscure potential advantages that may not be obvious since the links between a cohesive philosophical framework and the conceptual modelling approach adopted are implicit and unexamined. Identification of such links could bring benefit to the conceptual modelling community and consequently help to increase the quality of future information systems development.

In this paper, we have tried to bring to the attention of the information systems modelling community some of those arcane philosophical underpinnings. We have noted that there is no 'holy grail' i.e. there is no *one* philosophical suite of ideas that we can insist *should* be adopted. Rather, we take a more pragmatic (but still formal) approach in recommending consideration of whether or not any particular philosophy is both useful and implementable i.e. whether it is indeed understandable and usable to the benefit of the information systems community.

We have noted, in particular, philosophy-based concerns about commonly used modelling terms such as 'concepts' and 'properties/attributes', whilst noting also the value of incorporating ontological thinking and theories of language use.

Our final conclusion is more along the lines of 'caveat emptor' – modellers need to be aware of the philosophical history, albeit indecisive, that may more explicitly be incorporated into current and emerging approaches to modern conceptual modelling and information systems development.

References

1. Henderson-Sellers, B., Gonzalez-Perez, C.: Uses and abuses of the stereotype mechanism in UML 1.x and 2.0. In: Nierstrasz, O., Whittle, J., Harel, D., Reggio, G. (eds.) MoDELS 2006. LNCS, vol. 4199, pp. 16–26. Springer, Heidelberg (2006)
2. Atkinson, C., Kühne, T., Henderson-Sellers, B.: Systematic stereotype usage. Software and System Modelling 2(3), 153–163 (2003)

3. Locke, J.: An essay concerning human understanding, Digireads, Kansas (1690/2009)
4. Smith, B.: Beyond concepts: ontology as reality representation. In: Varzi, A., Vieu, L. (eds.) Procs. FOIS 2004, pp. 73–84. IOS Press (2004)
5. Partridge, C.: Business Objects: Re-Engineering for Re-use. Butterworth Heineman, Oxford (1996)
6. Lakoff, G.: Women, Fire, and Dangerous Things. What Categories Reveal about the Mind. Univ. Chicago Press (1987)
7. Stanford Encyclopedia of Philosophy (last major revision) (2011), `http://plato.stanford.edu/entries/concepts/` (accessed September 3, 2012)
8. Margolis, E., Laurence, S. (eds.): Concepts: Core Readings. MIT Press (1999)
9. Locke, J.: An Essay Concerning Human Understanding. Oxford Univ. Press, Oxford (1690, 1975)
10. Hume, D.: A Treatise of Human Nature. Oxford University Press, Oxford (1739, 1978)
11. Wittgenstein, L.: Philosophical Investigations. Macmillan, New York (1953)
12. Peacocke, C.: A Study of Concepts. MIT Press, Cambridge (1992)
13. Gruber, T.R.: A translation approach to portable ontologies. Knowledge Acquisition 5, 199–220 (1993)
14. Eriksson, O., Ågerfalk, P.J.: Rethinking the meaning of identifiers in information infrastructures. J. Assoc. Info. Systems 11(8), 433–454 (2010)
15. Henderson-Sellers, B.: On the Mathematics of Modelling, Metamodelling, Ontologies and Modelling Languages, Springer Briefs in Computer Science. Springer (2012)
16. Henderson-Sellers, B., Eriksson, O., Gonzalez-Perez, C., Ågerfalk, P.J.: Ptolemaic metamodelling? The need for a paradigm shift. In: Cueva Lovelle, J.M., et al. (eds.) Progressions and Innovations in Model-Driven Software Engineering. IGI Global (2013)
17. Shan, L., Zhu, H.: Unifying the semantics of models and meta-models in the multi-layered UML meta-modelling hierarchy. Int. J. Softw. Informat. 6(2), 163–200 (2012)
18. Bunge, M.: Treatise on Basic Philosophy: Ontology I: The Furniture of the World, vol. 3. Reidel, Boston (1977)
19. Wand, Y.: Ontology as a foundation for meta-modelling and method engineering. Inf. Software Technol. 38, 281–287 (1996)
20. Opdahl, A., Henderson-Sellers, B.: A template for defining enterprise modelling constructs. Journal of Database Management 15(2), 39–73 (2004)
21. Mellor, D.H., Oliver, A. (eds.): Properties. Oxford Univ. Press (1997)
22. Partridge, C.: Personal communication (2012)
23. Searle, J.R.: Speech Acts. An Essay in the Philosophy of Language. Cambridge Univ. Press (1969)
24. Guizzardi, G., Wagner, G.: Towards Ontological Foundations for Agent Modelling Concepts Using the Unified Fundational Ontology (UFO). In: Bresciani, P., Giorgini, P., Henderson-Sellers, B., Low, G., Winikoff, M. (eds.) AOIS 2004. LNCS (LNAI), vol. 3508, pp. 110–124. Springer, Heidelberg (2005)
25. Lowe, E.J.: The Four-Category Ontology: A Metaphysical Foundation for Natural Science. Oxford Univ. Press (2006)
26. Descartes, R.: Principia philosophiae (Principles of Philosophy) (1644)
27. Guizzardi, G., Masolo, C., Borgo, S.: In defense of a trope-based ontology for conceptual modeling: An example with the foundations of attributes, weak entities and datatypes. In: Embley, D.W., Olivé, A., Ram, S. (eds.) ER 2006. LNCS, vol. 4215, pp. 112–125. Springer, Heidelberg (2006)
28. Wand, Y., Weber, R.: An ontological analysis of some fundamental information systems concepts. In: Proceedings of the Ninth International Conference on Information Systems, Minneapolis (1988)

29. Wand, Y., Weber, R.: An ontological model of an information system. IEEE Trans. Software Eng. 16(11), 1282–1292 (1990)
30. Wand, Y., Weber, R.: On the deep structure of information systems. Information Systems Journal 5, 203–223 (1995)
31. Baldinger, K.: Semantic Theory: Towards a Modern Semantics. Palgrave Macmillan (1980)
32. Sowa, J.F.: Ontology, metadata, and semiotics, Conceptual Structures: Logical, Linguistic, and Computational Issues. In: Ganter, B., Mineau, G.W. (eds.) ICCS 2000. LNCS, vol. 1867, pp. 55–81. Springer, Heidelberg (2000)
33. Guizzardi, G.: Ontological foundations for structural conceptual models. CTIT PhD Thesis Series, No. 05-74, Enschede, The Netherlands (2005)
34. Ogden, C.K., Richards, I.A.: The Meaning of Meaning. Harcourt, Brace and World, New York (1923)
35. Peirce, C.S.: Collected Papers of C. S. Peirce. In: Hartshorne, C., Weiss, P., Burks, A. (eds.) 8 vols. Harvard University Press, Cambridge (1958)
36. Ullmann, S.: Semantics: An Introduction to the Science of Meaning. Basil Blackwell, Oxford (1964)
37. Jackson, M.: Some Notes on Models and Modelling. In: Borgida, A.T., Chaudhri, V.K., Giorgini, P., Yu, E.S. (eds.) Mylopoulos Festschrift. LNCS, vol. 5600, pp. 68–81. Springer, Heidelberg (2009)
38. Wilmont, I., Brinkkemper, S., van de Weerd, I., Hoppenbrouwers, S.: Exploring intuitive modelling behaviour. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Ukor, R., et al. (eds.) BPMDS 2010 and EMMSAD 2010. LNBIP, vol. 50, pp. 301–313. Springer, Heidelberg (2010)
39. Gonzalez-Perez, C., Henderson-Sellers, B.: Modelling software development methodologies: a conceptual foundation. J. Sys. Softw. 80(11), 1778–1796 (2007)
40. Lorenzatti, A., Abel, M., Fiorini, S.R., Bernardes, A.K., dos Santos Scherer, C.M.: Ontological primitives for visual knowledge. In: da Rocha Costa, A.C., Vicari, R.M., Tonidandel, F. (eds.) SBIA 2010. LNCS, vol. 6404, pp. 1–10. Springer, Heidelberg (2010)
41. Lorenzatti, A., Santin, C.E., Paesi da Silva, O., Abel, M.: A representation frame-work for visual knowledge. In: Vieira, R., et al. (eds.) Proceedings of Joint IV Seminar on Ontology Research in Brazil and VI International Workshop on Metamodels, Ontologies and Semantic Technologies, CEUR Workshop Procs., vol. 776, pp. 177–182 (2011)
42. Austin, J.L.: How To Do Things With Words. Oxford Univ. Press (1962)
43. Frege, G.: Über Begriff und Gegenstand (On Concept and Object). In: Vierteljahrsschrift für wissenschaftliche Philosophie, vol. XVI, pp. 192–205. Fues's Verlag (1892) (transl. Geach and Black, 1952)
44. OMG: OMG Unified Modeling Language™ (OMG UML), Infrastructure Version 2.4.1, OMG Document Number: formal/2011-08-05, 230pp (2011)
45. Lilly, L.: Use case pitfalls: top 10 problems from real projects using use cases. In: Firesmith, D., et al. (eds.) Procs. TOOLS 30, pp. 174–183. IEEE Comp. Soc. Press (1999)
46. Mylopoulos, J.: Conceptual modeling and Telos. In: Loucopoulos, P., Zicari, R. (eds.) Conceptual Modeling, Databases and CASE, ch. 4, pp. 49–68. Wiley (1992)
47. Kaschek, R., Delcambre, L. (eds.): The Evolution of Conceptual Modeling. From a Historical Perspective towards the Future of Conceptual Modeling. LNCS, vol. 6520. Springer, Heidelberg (2011)
48. Incipit: ConML Technical Specification. ConML 1.3 (2013), `http://www.conml.org/Resources_TechSpec.aspx`
49. Gonzalez-Perez, C.: A conceptual modelling language for the humanities and social sciences. In: Rolland, C., et al. (eds.) Sixth International Conference on Research Challenges in Information Science (RCIS 2012), pp. 396–401. IEEE Computer Society (2012)

Towards a Sociomaterial Ontology

Maria Bergholtz[1], Owen Eriksson[2], and Paul Johannesson[1]

[1] Department of Computer and Systems Sciences, Stockholm University
{maria,pajo}@dsv.su.se
[2] Department of Informatics and Media Uppsala University
owen.eriksson@im.uu.se

Abstract. The management of social phenomena in conceptual modelling requires a novel understanding of the notion of representation. In particular, the principles for the existence and identification of objects need to be reconsidered. To do this, the paper draws on the current ontological discourse in information systems engineering and proposes a sociomaterial ontology for supporting conceptual modeling. The ontology shows how organisational entities are grounded in physical ones and how they can be understood in terms of deontic notions like privileges, duties and powers. The sociomaterial ontology is able to assist designers in creating understandable and robust conceptual models.

1 Introduction

Work on ontologies for information systems design reflects a distinction between physical and social phenomena. Much of this work has its origins in ontologies for physical things and phenomena, such as the BWW-ontology [15]. In order to address social aspects, some authors have proposed socially oriented ontologies, such the Enterprise Ontology by [1]. These ontologies focus on social constructions like agents, organizations, rights, obligations, commitments, contracts, and social actions. An ontology that addresses physical as well as social aspects is UFO proposed by [4]. While these ontologies have resulted in a deeper understanding of the subject matter of conceptual modeling and information systems, there is still a need for further clarifying the relationships between the physical and social domains, as suggested by recent work on sociomateriality, [9].

The main goal of the paper is to propose a sociomaterial ontology for conceptual modeling. Section 2 elaborates on the how the material and social world can be represented by language and presents the sociomaterial ontology, which is primarily based on speech act theory, [10], universal pragmatics (formal pragmatics) [5], rights analysis, [7], and social ontology, [2, 11]. Section 3 illustrates how the ontology can be used for supporting conceptual modelling. Finally, Section 4 concludes the paper and suggests directions for further research.

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 341–348, 2013.

2 A Sociomaterial Ontology

2.1 The Construction of Social Reality

For the purpose of this paper, a first distinction is made between physical things and humans. Humans, as well as other biological beings, are grounded in physical things in the sense that they are existence dependent on them through an asymmetric, transitive and down linear grounding relationship, as suggested in [8].

Humans can perform physical actions, i.e. create, modify and destroy physical things. We here take the view that also some physical things can perform physical actions, though not intentionally, e.g. an ATM ejecting a credit card. The notions introduced above are shown in the lower part of Fig. 1 as the classes *Material Entity, Physical Entity, Human Being, Physical Media*, and *Physical Action*.

Humans regularly interact with each other and are related in various ways. Groups of people and their relationships may become more or less stable over time, resulting in social structures. A *social structure* can be defined as a group of people who are related to each other by holding some beliefs or dispositions with regard to some practices, [2]. In order to describe these kinds of structures, the notion of institution is helpful. Institutions can be defined as "systems of established and prevalent social rules that structure social interactions", [6].

In order to clarify the basis of social phenomena and how the world is constituted by the use of language, we will here follow the work of John Searle [10, 11]. A key assumption is that the social world consists of social relationships that bundle rights, and some of these have been created according to formal institutions.

2.2 Rights and Institutional Entities

A key observation is that not only humans but also other kinds of entities may have rights. For example, organisations may have duties and privileges to other organisations as well as to people. In order to capture the idea that some objects may have rights, the notion of institutional entity is introduced. An *institutional entity* is an object that may have rights within some institution, where a right is defined according to W. N. Hohfeld, [7]. An institutional entity has a *claim* on another institutional entity if the second entity is required to act in a certain way for the benefit of the first entity, typically by carrying out some action. Conversely, the second entity is said to have a *duty* to the first one. An institutional entity has a *privilege* on an action if she is free to carry out that action in accordance with the regulations of an institution. A *power* is the ability of an institutional entity to create or modify a social relationship. An example is that a person owning a piece of land has the power to sell it to someone else, thereby creating a new ownership relationship for that piece of land. In our work, an institutional entity has *active* power if it can carry out an action that contributes to the creation or modification of a social relationship. An institutional entity has *passive* power if its presence in a certain situation contributes to the creation or modification of a social relationship.

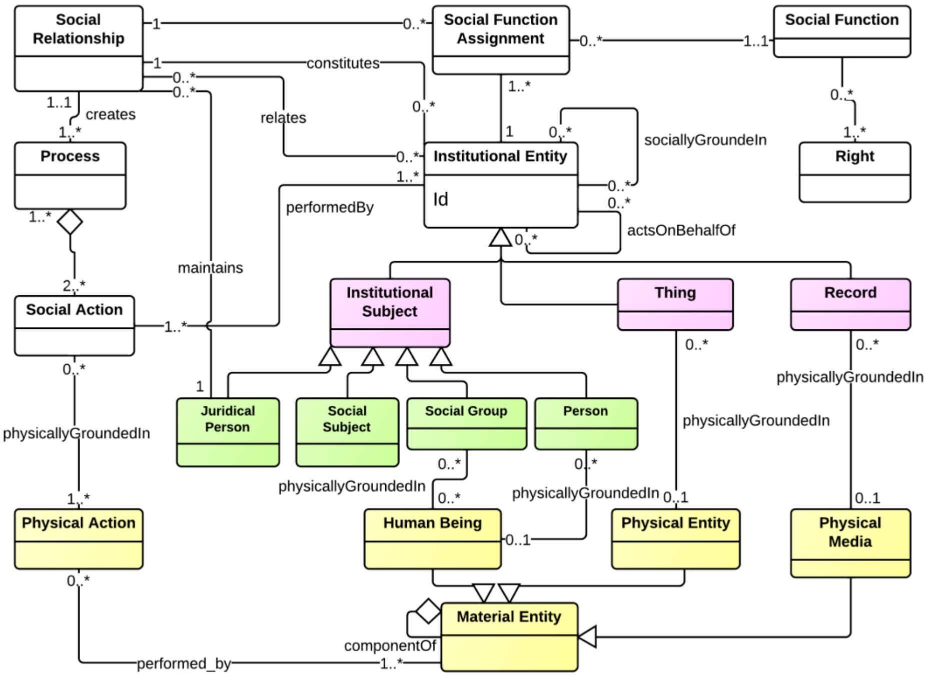


Fig. 1. The socio-material ontology

Typically, rights do not occur in isolation but are bundled together in social relationships that bind several institutional entities together. A *social relationship* is a collection of rights among two or more institutional entities that are defined according to the rules of some institution. An example of a social relationship is a contract between a customer and a supplier for some products. The contract would primarily consist of a number of duties for the supplier to deliver products and for the customer to pay for them. Institutional entities always exist within social relationships, i.e. every institutional entity is existence dependent on a social relationship. For example, a customer exists only if there is a social relationship between the customer and a supplier; if the social relationship ceases to exist, so does the customer. In other words, the social relationship *constitutes* the customer - the customer exists because of the social relationship. Thus, some social relationships constitute new institutional entities, while others only *relate* already existing institutional entities, see Fig 1.

When an institutional entity has been created, it will receive a number of rights, and during its lifetime it will continue receiving and losing rights. In order to represent these changes of rights, we use the notion of social functions where a *social function* is as a set of kinds of rights, e.g. those of a husband in a Marriage relationship. Social functions can be assigned to institutional entities within a social relationship, as modelled by the class *Social Function Assignment* in Fig. 1. Through such an assignment, an institutional entity will hold the rights given by the social function.

2.3 Kinds of Institutional Entities

There exists a large variety of institutional entities, ranging from companies and citizens to products and passwords. The taxonomy suggested here is based on the grounding, [8], of institutional entities as well as the kinds of rights they may possess. On the top level, we identify three kinds of institutional entities: institutional subjects, things, and records. An institutional subject is directly or indirectly grounded in a human being, while a thing is grounded in a physical object and a record in physical media.

Every institutional entity is directly or indirectly related to one or more material entities. Some entities are physically grounded in material entities through the relationship *physically grounded*, cf. [8]. An institutional entity may also be related to institutional entities through the relationship *socially grounded*, meaning that when the first entity is created it is based on the other entity. For example, when a credit card is issued and a credit card customer is created, it typically has its origin in a citizen of some nation. Social grounding, unlike physical grounding, does not imply existence dependency, it is, however, still an asymmetric, transitive and down linear relationship.

An *institutional subject* is an institutional entity that is existence dependent on one or more human beings. Four kinds of specializations of institutional subjects can be identified based on an analysis of (i) what types of rights an institutional subject can possess, in particular if it can have duties, i.e. be responsible to other subjects, (ii) whether or not an institutional subject is grounded.

A *person* is an institutional entity, physically grounded in a single human being or socially grounded in another person that can have duties as well as privileges and powers. Examples of persons are citizens, students, and customers. A *juridical person* is an institutional entity for which there always must exist at least one person that can perform actions on its behalf. A juridical person can have duties as well as privileges and powers. Examples of juridical persons are companies and associations. A *social group* is an institutional entity physically grounded in a group of human beings or socially grounded in a group of persons that can have passive powers but neither duties, privileges nor active powers. An example of a social group is a household. The household itself does not have any duties but it can have passive powers, e.g. a household with children may bestow a right on one of the parents to receive a monthly allowance. A *social subject* is an institutional entity for which there always must exist at least one person that can perform actions on its behalf. A social subject can have passive powers but not duties, privileges or active powers. In contrast to a social group, the social subject is not existence dependent on the persons that can act on its behalf. An example of a social subject is a rock band, for instance "The Beatles". A *thing* is an institutional entity, physically grounded in a single physical entity or socially grounded in another thing that can have active and passive powers, as well as privileges but not duties. Examples of things are traffic lights and coins. A *record* is an institutional entity that is physically grounded in physical media or socially grounded in other records, e.g. a password or a discount code. Records cannot have duties or privileges but only passive powers. For example, a password can be used to provide privileges to a document.

Humans as well as physical things are able to carry out physical actions. Physical actions always change physical entities, but they can sometimes also be used for creating, modifying and terminating social relationships. In such cases, people perform social actions through physical actions, e.g. entering a treaty can be done by signing a document. A social relationship is created, modified, and terminated through a process of such social actions.

3 Using the SMO - Case and Comparisons

We introduce a running case to illustrate applications of the socio-material ontology and to compare it to a well-known upper-level ontology, UFO [4]. In Sweden, the right to financial aid is governed by the SoL, Socialtjänstlagen (Social Services Act) [12], which specifies rights of municipalities and residents. Municipalities have the power to assign allowances (a right) to household members depending on the number of people living in the household, income, etc. A household is a number of people who live in the same dwelling; they can be biologically related but this is not necessary. In order to make decisions on the right to allowances, the municipality has to determine the members of a household. In terms of the sociomaterial ontology, the municipality creates a household relationship - a social relationship that constitutes a household and a number of household members. In doing so, it consults information from another juridical person, the Swedish national housing office that records what people are residents on what addresses. In order to be constituted as a member of a household, you must have been constituted as a resident by the Swedish national housing office (a juridical person outside of the municipality). A household member can be either an adult or a child. When a household is created, its adults are given duties with respect to its children, and the latter are given corresponding claims with respect to the adults. One of the adults is, optionally, granted rights to allowance from the municipality if the household does not reach an income-level specified in the SoL. The household itself (a social group) is not granted any rights other than passive powers, i.e. it is only in the presence of the household that its members can be granted rights.

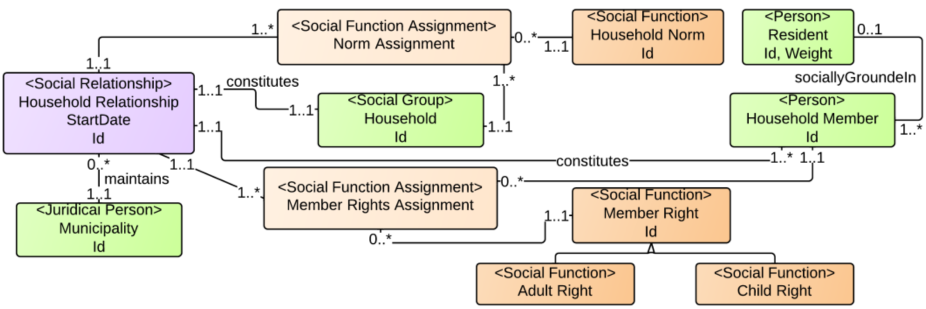


Fig. 2. Social functions (bundled rights) assigned to household members and households

The SMO can be used to guide conceptual modellers when designing models for social domains, in particular in inter-organizational contexts. The main guideline is that every class introduced in a domain model should be related to one of the SMO classes, e.g. by stereotyping. Applying the SMO for the running case results in the model in Fig. 2.

Next, the running case is modelled using the UFO ontology [4]. Using the UFO, adults or children in a household are represented as phases. A *Phase* is a an *anti-rigid sortal*, where sortal refers to a core category in UFO representing entity types that carry a principle of identity for its instances, i.e. a principle for determining whether two instances are the same or not. The statement that a phase is anti-rigid means that an instance of the phase is not an instance of it in all possible worlds. Class *Resident* is a specialization of *Person* - a *kind* from which *Resident* inherits its principle of identity. The fact that a member of a household has to be a resident is modelled as *Household Member* being a specialization of *Resident*, both of which are roles. A *Role*, like a *Phase*, is anti-rigid, e.g. persons who take on the role of resident or household member may move in and out of these roles. Roles differ from phases with respect to their specialization conditions, where phases depend on intrinsic properties of the phased entity, while roles depend on extrinsic (relational) properties of the role. For example, what determines whether or not a resident should be specialized into an *Adult* or a *Child* is the intrinsic property 'age'. If a resident is classified as household member or not depends on the relations the person has to the other potential members of the household, in particular whether they are co-located on the same address. *Household*, finally, is a *Collective*, which typically provides an extensional principle of identity, i.e. two collectives are the same if their members are the same.

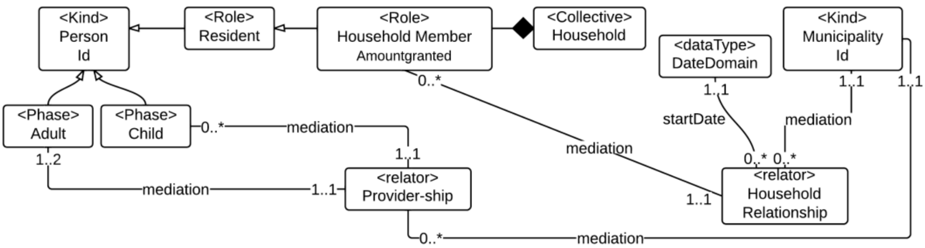


Fig. 3. Running case modelled using UFO [Guizzardi05]

Comparing Fig 2 and 3, one distinguishing feature of the SMO is how generalization relationships are handled, e.g. *Household Member* is not viewed as a subclass of *Person* in the SMO. The ontological reason for this is that household members, children and adults, are constituted by the municipality, while persons are constituted by the Swedish state (a Juridical Person). For the same reason, *Household Membe*r is not a subclass of *Resident*, since instances of the latter are constituted not by the municipality but by the Swedish national housing office. In contrast, in the UFO model, *Household Member* is a subclass of *Resident*. The SMO model instead only relates *Household Member* to *Resident* or *Person* such that a household member is socially grounded in a resident. Thus, a household member is viewed as different and not existence dependent, though related to, a person. One advantage of this approach is that we get a looser coupling between the classes *Household Member* and *Person*, which makes the model more robust to changes. One scenario is that the Swedish national housing office changes the principle of application of *Person* (or *Resident)* so that it becomes more restricted. This could invalidate the subclass relationship between *Household Member* and *Person/Resident*, as there may be household members that are no longer residents. To handle this, the subclass hierarchy in UFO would need to change as the class *Resident* is replaced. In the SMO, the change is more restricted, as we only would need to replace *Resident* without making any changes to the subclass hierarchy. Furthermore, in SMO the members of a household are given different rights depending on the social function assigned to them - either duty to provide (adults) or claims to be provided (children). In UFO, the phases *Child* and *Adult* are not characterized by the rights the household members receive in the household but on intrinsic properties of a person, in this case his or her age. The class *Household*, finally, is a *Social Group* corresponding closely to the UFO concept *Collective* (here a group of persons), i.e. a household is existence dependent on its members.

4 Concluding Remarks

The main contribution of this paper is a sociomaterial ontology to be used for supporting conceptual modelling. Using the SMO as guidance in designing conceptual models will provide advantages in terms of understandability and robustness. The ontology identifies and distinguishes key notions for modelling social phenomena and associating them to material ones. In particular, the models will explicitly and systematically represent the rights that govern the social interaction in the domain, typically in the form of attributes included in classes representing social relationships or institutional entities. Material aspects will also be made explicit in a structured way, typically by means of attributes included in classes representing institutional entities. Furthermore, the ontology supports the creation of robust models by using grounding relationships instead of generalization relationships, thereby making models more resilient to changes in identifier or principle of application, as discussed in Section 3.

The SMO has been compared to the UFO ontology though an example. Another work similar to SMO is the Agent-Object-Relationship metamodel (AOR) by [14]. AOR also makes use of deontic notions for analysing and explicating social phenomena. However, the SMO differs from AOR in several respects. While AOR classifies agents into biological, institutional and artificial ones, SMO instead shows how institutional subjects can emerge from other entities through grounding relationships. This enables us to define several institutional subjects grounded in, for example, the same human being. Furthermore, the SMO allows powers to be assigned also to things, i.e. entities grounded in material objects. Another difference is that we have made social relationships explicit as containers for claims and other rights, which allows for a convenient representation of contract terms. In this respect, our ontology is similar to the REA ontology, [3], which distinguishes between contracts and commitments.

Further work on the sociomaterial ontology will include consolidation, in particular how to manage the notion of social function on various levels of abstraction, which is closely related to issues in role modelling, [13].

Another issue for further research is to consolidate the categorization of institutional subjects. For example, in Section 2, it was stated that neither social groups nor social subjects can have duties (as neither can be held legally responsible). In symmetry with this, it was stated that they cannot have claims. However, it might be argued that not having duties does not necessarily entail not having claims. If so, the household in Section 3 could actually have claims on the municipality for getting allowances. Further analysis of the relationship between rights and the categories of institutional entities needs to be undertaken.

References

1. Dietz, J.L.G.: Enterprise Ontology – theory and methodology. Springer, Heidelberg (2005)
2. Elder-Vass, D.: The causal power of social structures: emergence, structure and agency. Cambridge University Press (2010)
3. Geerts, G., McCarthy, W.E.: An Accounting Object Infrastructure For Knowledge-Based Enterprise Models. IEEE Int. Systems & Their Applications, 89–94 (1999)
4. Guizzardi, G.: Ontological Foundations for Structural Conceptual Models, CTIT PhD Thesis Series, No. 05-74, Telematica Instituut Fundamental Research Series, No. 015, TI/FRS/015 (2005)
5. Habermas, J.: What is Universal Pragmatics? In: Communication and the Evolution of Society. Heinemann, London (1979)
6. Hodgson, G.M.: What Are Institutions? Journal of Economic Issues XL (1) (2006)
7. Hohfeld, W.N.: Fundamental Legal Conceptions. In: Corbin (ed.) Green-wood Press, Westport (1978)
8. Masolo, C.: Levels for Conceptual Modeling. In: De Troyer, O., Bauzer Medeiros, C., Billen, R., Hallot, P., Simitsis, A., Van Mingroot, H. (eds.) ER Workshops 2011. LNCS, vol. 6999, pp. 173–182. Springer, Heidelberg (2011)
9. Orlikowski, W., Scott, S.: Sociomateriality: Challenging the Separation of Technology, Work and Organization. The Academy of Management Annals 2(1) (2008)
10. Searle, J.: Speech acts: an essay in the philosophy of language. Cambridge University Press (1969)
11. Searle, J.: Social Ontology Some Basic Principles. Anthropological Theory 6, 12–29 (2006)
12. SoL, `http://www.riksdagen.se/sv/Dokument-Lagar/Lagar/Svenskforfattningssamling/Socialtjanstlag-2001453_sfs-2001-453/`
13. Steimann, F.: On the representation of roles in object-oriented and conceptual modeling. Data & Knowledge Engineering 35, 83–106 (2000)
14. Wagner, G.: The Agent–Object-Relationship metamodel: towards a unified view of state and behavior. Information Systems 28(5), 475–504 (2003)
15. Wand, Y., Weber, R.: On the deep structure of information systems. Information Systems Journal 5(3), 203–223 (1995)

Knowledge Organization and the Conceptual Basis for Building Classification Systems for Complex Documents: An Application on the Brazilian Popular Song Domain

Rodrigo De Santis

Information Science Postgraduate Program, Brazilian Institute of Information Science and Technology (IBICT) and Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil
rodrigo@aquelamusica.com.br

Abstract. Knowledge Organization (KO) is one of the main activities of the Information Science field.

The theoretical and epistemological inputs from Information Science have proved themselves crucial for the construction of new classification systems – the applied dimension of KO – which take into account the multidisciplinarity inherent to complex documents and which are capable of expressing their multidimensional nature.

This article discusses the theoretical grounding for the classification of popular songs and also presents the construction of an ontology-based system for this kind of complex document.

Keywords: Knowledge Organization, classification, ontology, popular songs, ontology-based system.

1 Introduction

The classification of complex artistic documents – such as the popular song – presents major challenges to the Knowledge Organization (KO) field. The main difficulties are related to consistently accounting for the multidisciplinary aspects and the multidimensionality of this kind of document.

The multidisciplinarity refers to the contributions of different areas that help in constructing the knowledge contained in an artistic document, taking into account its historical, social and cultural context [1].

The multidimensionality refers to the existence of several layers of meaning within the same document. While the classification may never be fully free from choices that will favor a certain aspect of the document to the detriment of other possible aspects, it should be considered from a consistent theoretical framework. The consistency will be as great as the ability of the classification to consider the different dimensions of the documents, in order to allow "perspective" views of knowledge [2].

From the relationship between the multidisciplinarity and multidimensionality arises the third challenge for the classification of artistic documents: the polysemy. Because it involves different disciplines and due to the fact that it can be appreciated

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 349–360, 2013.

from different perspectives, a complex artistic document has multiple meanings and may yield different "meaning effects"[1] depending on how it is used.

The research on Knowledge Organization, within the realm of Information Science, seeks to address these challenges, collaborating in the investigation of an epistemological grounding that may work as a basis for the classification of complex documents, such as popular songs.

The establishment of these epistemological groundings subsidizes the construction of classification systems – applied dimension of KO – able to respond to these challenges.

This paper presents the problems of classification of popular songs and shows the first results about the definition of an epistemological grounding specific to the field of popular songs and its application through building an ontology-based classification system.

2 Classification of Popular Songs

2.1 The Popular Song as a Complex Document

The recognition of the popular song as an object of scientific study is recent. Until the early 1990s, the popular song had only been partially studied, inasmuch as the lyrics were sometimes taken as literary and historic objects, and sometimes the musical elements were considered apart from the lyrics [4].

This way of thinking, i.e., "subdivided, compartmentalized, monodisciplinary, quantifier" is identified by Morin as opposite to the "true knowledge" or "act of knowing" that must be based on contextualization, in an "unbroken spiral", in which there are movements of "separating in order to analyze, and of reconnecting in order to synthetize or render complex" [5].

Only after it started to be considered as a syncretic object and no longer "poetry set to music" or "music over a literary text" did the popular song reach the status of a language on its own and had its multidisciplinarity appropriately understood [6].

The Brazilian popular song – object of the research presented in this paper – is presented as an object which is also relevant outside the context of Brazilian culture, not only for its international influence, but mainly because it confirms the characteristics of multidisciplinarity and multidimensionality that are common to the popular songs made in other parts of the world [4].

2.2 Classification by Descriptive Metadata and by Musical Genre

Although presently the popular song is already understood as an autonomous document, carrier of its own language, it was and continues to be classified mostly based on some descriptive metadata (such as title, name of songwriter, date of

---

[1] "Meaning effect" is a concept borrowed from psychoanalysis to describe the effect produced by the meaning, not the meaning by itself [3].

recording etc.) and on a category inherited from traditional musicology: the musical genre.

The genres of classical music originated from the grouping of several elements, such as the song structure, its instrumentation, or its social function (e.g.: sonata, concerto, and mass, respectively). In classical music, the genre-defining elements are considerably stable, which makes it possible to arrange them in a fairly precise fashion in order to coherently define a musical genre[2].

Initially, "popular song" was one more taxonomy item of genres used by classical music, which already possessed a genre "song" to designate classical vocal works which imparted "musicality" to poetry [7]. However, due to the heterogeneity of popular song manifestations, the breakdown of the "popular song" genre soon became necessary, seeking to accommodate the variations being identified. Yet, contrary to what happens in classical music, in popular song the constitutive elements are inherently unstable [8], and, in practice, musical genres inform little about its structure, instrumentation, or social function [9].

The Mass Culture Industries have belatedly recognized popular music as a commercial product [10]. For several years, popular songs were, in most parts of the West, relegated to a position of inferiority compared to classical music [11]. Nevertheless, once its role over popular music had been established, the Mass Culture Industries availed of the assortment into musical genres used, until then, in an ad hoc manner by songwriters and performers, who classified their songs freely, and sought to simplify it, redefining the genre taxonomy into a more limited and objective formulation. The goal was to classify the consumer public, and not the works, thus decreasing the risks of its commercial operation [12]. This redefinition of musical genres for popular songs gave rise to denominations that expresses little or nothing about the contents to which they were associated for marketing reasons, as is the case of "World Music" or "Latin Music".

Internally, popular songs are composed of a great range of elements, such as, the rhythmic, harmonic and melodic components; the themes addressed in the lyrics; the narrative structures with its projections of person, time, and space [13]; the interpretative nuances; the variations deriving from the arrangement; the modalizations involved in the discourse; their historical and social conjuncture etc.

The construction of the meaning effects produced by the songs occurs through mixtures of their constitutive elements [8]. These mixtures, however, do not obey pre-established rules, which generates the typical instability of popular songs and increases the complexity of the multiple dimensions contained therein. All of this indicates, therefore, that the classification of songs based on a set of descriptive metadata and on the category of musical genre is, at best, insufficient. For a deeper investigation concerning the existent knowledge in a popular song and the possible ways to capture and organize this knowledge through an efficient classification system, it is imperative to adopt techniques based on a multidisciplinary approach.

[2] Even in classical music, the mentioned stability no longer exists since the twentieth century with the changes brought by contemporary music, serial music and electronic music.

3 Popular Songs Classification Systems

3.1 Paradigmatic Changes in the Classification of Songs

Despite its late recognition, the popular music market quickly proved to be promising and profitable, which led record labels and other players of the phonographic industry to invest heavily in mechanisms which would allow them to gather even more consumers interested in the same type of music, and consequently increase the record sale [14]. By the end of the 1970s, when the first classification systems emerged with the goal of recommending products to users based on similarity criteria – the so called Recommender Systems – the first object was popular music [15][3].

With the advent of the web and the transformations of the consumer market that took place by the end of the twentieth century, popular music was once again one of the main exponents of the new order that was being established. With the decrease in record sales and the increase of digital reproduction mechanisms, phonographic industries began abandoning the conception of music as a product to think of it as information and service [12].

This paradigmatic shift has fundamental implications in the conception of the popular song as a knowledge object inside classification systems. If it was already important to consider the song as a complex object from the artistic point of view, this perspective also derives support from the commercial standpoint, reinforcing the need to build classification systems which take into consideration the multiple dimensions contained in the songs.

3.2 Classification System Models

The classification schemas used for music recommendation are grouped in three models according to their type of operation: content-based, collaborative and hybrid [17] [18] [19].

The content-based ones consider the descriptions of objects (their descriptive metadata) or the results of analyzes of the audio content or score (frequencies, keys, pitches, durations, etc.).

The collaborative ones consider the opinions and classifications made by other users of the system, identifying the objects that fit into similar use profiles.

The hybrid ones, as the name suggests, combine the collaborative method with the content-based method.

Aspects of the collaborative model

Last.fm [20] – an important web music recommender system – adopts the hybrid model. In the content-based aspect, Last.fm categorizes music through a conventional set of metadata (title, album, songwriter, performer and recording date); in the collaborative aspect it allows users to classify songs using personalized tags to label

---

[3] For a historical view of the development of the classification systems of music recommendation, it is recommended to read ADOMAVICIUS & TUZHILIN [16].

artists and songs, also creating communities to group profiles that have like musical interests.

Several works have been dedicated to analyzing the use of tags in collaborative systems of popular music classification [11] [12] [21] [22].

These studies reveal that the number of tags created by users in collaborative models tends to be significantly large, since the adoption of a non-controlled vocabulary ends up multiplying the uses and ends attributed to classifications [23].

Another finding of the studies regarding the use of tags in collaborative systems points towards the practical difficulty in using tags at the moment of information retrieval. Because they are not hierarchized, tags created by users often are not understood by other users. And, despite the amount of tags grows indefinitely, there is a tendency of concentrating the use of tags for a smaller set of items. In practice, the most popular items receive many identical classifications while most of the items receives no classification at all, which leads to a tendency of clustering of classifications [23].

Importance of the content-based model

The onset of classification systems based on collaborative models in the web has caused an initial enthusiasm resulting from the expectation that it would be possible to promote the democratization of access to musical content. However, although it does promote indisputable advances concerning the manifestation of users' opinions, as could be seen, the classification based on collaborative methods is not able to solve the problem of massification and unbalance of classifications which, in the end, are applied to some few items that fit in hegemonic categories.

From these findings, the content-based aspect has gained new thrust. A confirmation of this is the concern manifested in one of the main study groups related to musical information, the ISMIR - The International Society for Music Information Retrieval [24]. In 2009, one of its founders, J. Stephen Downie, expressed the group's concern for increasing the number of researches that consider the symbolic and metadata aspects of musical objects [25].

The problem, however, lies, once again, in the lack of epistemological grounding for the definition of categories which are both effective and suitable for the analysis and resulting classification of existent content in popular songs.

3.3 Discourse Formation as Conceptual Basis for Content Analysis

A possible perspective, offered within the realm of Social Sciences, for determining a conceptual basis destined for the analysis of knowledge existent in complex documents is the concept of "discourse formation" proposed by Foucault.

According to Foucault, the rules that constitute discourse as a coherent unit are always presented from a system of relations. These would be the relations between the constitutive elements of the objects, apparently disperse in discourse, which make it possible for regularities and recurrences to be identified. These relations present themselves in a system of dependencies that can be organized hierarchically. Foucault believes, therefore, that inasmuch as relations form regularities amidst apparent dispersion, they refer to a system of specific rules which, in turn, govern discourse formation [26].

Table 1. Rules and elements identified in popular songs

Element	Content type	Fill type	Related elements
Title	Metadata	descriptive	
Songwriter	People domain	descriptive	
Creation date	Metadata	descriptive	
Destination	Taxonomy	descriptive	
Lyrics	Metadata	descriptive	
Language	Taxonomy	descriptive	
Person	Taxonomy	formation rule	Lyrics
Time	Taxonomy	formation rule	Lyrics
Space	Taxonomy	formation rule	Lyrics
Modalization	Taxonomy	formation rule	Person; Time; Space
Theme	Taxonomy	formation rule	Modalization; Destination
Emotion	Taxonomy	formation rule	Theme; Melodic Profile; Rhythmic profile
Narrative Structure	Taxonomy	formation rule	Modalization; Melodic Profile
Performer	People domain	descriptive	
Musician	People domain	descriptive	
Instrument	Taxonomy	descriptive	
Instrumentation	Taxonomy	formation rule	Instrument; Destination
Label	Metadata	descriptive	
Album	Metadata	descriptive	
Recording Date	Metadata	descriptive	
Recording place	Places domain	descriptive	
Musical genre	Taxonomy	descriptive	
Tessitura	Metadata	descriptive	
Tempo	Taxonomy	descriptive	
Mode	Taxonomy	descriptive	
Key	Taxonomy	descriptive	
Dynamics	Taxonomy	descriptive	
Rhythmic profile	Taxonomy	formation rule	Instrumentation; Mode; Key, Dynamics
Melodic profile	Taxonomy	formation rule	Tessitura, Tempo, Mode, Key
Harmonic profile	Taxonomy	formation rule	Melodic profile, Rhythmic profile

Deconstructing popular songs, using the analogy proposed by Foucault of the work of an “archaeologist of knowledge”, aims at the identification of the constitutive elements of a song, of how the rules are set from which knowledge is built, and of what are the systems of relations that manifest themselves recurrently.

The multiple aspects of knowledge involved in popular songs require different tools of analysis (for the melodic profiles, for the narrative and discoursive aspects, for the linguistic aspects, for the modeling of emotions and mood etc.).

By applying this multidisciplinary approach, recurrent rules and elements of Brazilian popular songs, summarized in table 1, were identified.

Although some of the identified elements are descriptive metadata already commonly employed in classifications (such as Title, Album, Year of recording etc.), this type of approach promotes the emerging of new categories originated from the recurrence of these elements (such as Modalization, and Instrumentation) and allows formation rules of to be identified, which govern the emerging of other inherently unstable categories, such as rhythmic, melodic, and harmonic profiles.

The construction of a classification system from the results obtained with this type of analytical approach needs to take into consideration, therefore, in addition to the terms and concepts involved, the complex system of dependency and relations between them. This type of representation is achieved through the use of thesauri or classification tables, but it has proven to be promising when presented in ontology, where it is possible to define explicitly each type of relation [28].

4 Application on an Ontology-Based Classification System

4.1 What's That Song?

The project "Aquela Música" (That Song) received this name due to its initial intention: to answer this routinely asked question when one tries to remember (or find out) a song about which there is little or fragmented information: "What's That Song? The one that talks about that subject, the one sang by that person and that has that other guy playing the flute", for example [29].

Departing from fragments of songs and rebuilding them from the identified relations and rules, according to the theory of discourse formation, an ontology-based system was elaborated and aims at organizing and classifying knowledge contained in popular songs and allowing its retrieval through an interactive interface[4].

4.2 Ontologies-Based System

The ontology of the Aquela Música project is written in RDF (Resource Description Framework). One of the concerns during its development was to make it compatible with the established patterns through the W3C consortium so, in the future, it would be possible to integrate it with the semantic web initiatives, such as the Linked Data Platform Working Group – whose goal is to define patterns which are reusable and applicable to the description of resources available in the web [30].

Also with the goal of associating itself to the currently ongoing actions, the ontology created for the Aquela Música project started at the integration of three already established ontologies, extending to accommodate the elements innate to the sphere of popular songs, as well as the rules identified from the effected analyses. The ontologies that serve as a basis for Aquela Música are: Music Ontology [31], Friend of a Friend Ontology (FOAF) [32], and COMUS Ontology (Context-based Music Recommendation) [33].

---

[4] The prototype can be used on the website `http://www.aquelamusica.com.br`

The Music Ontology

Designed specifically for the description of phonograms, artists and albums, the Music Ontology is an important part of the DBTune project, whose purpose is to semantically map, in RDF language, the records of phonograms and albums from the collections of MySpace, MusicBrainz and BBC radio websites [35].

The Music Ontology can be divided into three levels of expressiveness – from the simplest one to more complex ones. The first level aims at providing a vocabulary for editorial information (tracks/artists/releases etc). The second level aims at providing a vocabulary for expressing the music creation workflow (composition, arrangement, recording etc). The third level aims at providing a vocabulary for complex event decomposition, to express, for example, what happened during a particular performance, what is the melody line of a particular work etc. [37]

The first two levels of the Music Ontology are rather stable and have been used in the Aquela Música project to accommodate their descriptive metadata. The third level could not be availed since it is not capable of properly accommodating the data derived from the decomposition of the popular song through the discursive formation technique.

The FOAF Ontology

The FOAF ontology is used for describing people, their activities and relationships with other people and objects.

The FOAF is also divided in three levels. The first level - Core - describes characteristics of people and social groups that are independent of time and technology; as such, they can be used to describe basic information about people in present day, historical, cultural heritage and digital library contexts. The second level - Social Web - contains the terms that may be used when describing Internet accounts, address books and other Web-based activities. The third level, Linked Data utilities, is part of the effort of fitting the FOAF into the Linked Data Platform.

Within the Aquela Música project, only the first level has been used to describe the Songwriters, Performers and Musicians and the relationships between themselves and the musical piece.

The COMUS Ontology

The COMUS Ontology is the most specific domain-oriented ontology found for the music classification field. It is composed of 862 classes and 61 properties that reflect several aspects – from the basic elements which are contained in the musical pieces (such as mode, key and pitch) to some elements derived from analyses such as emotions and mood [33] [36].

Throughout the development of Aquela Música ontology it hasn’t been possible to access the sources of the COMUS Ontology. This ontology had to be rebuilt from the descriptions contained in the academic papers cited above. Thereby, the Aquela Música made use of the concepts defined by the COMUS Ontology project, but reproduced them internally, thus making them part of the core of the Aquela Música ontology itself.

4.3 Information System for Classification and Retrieval

The ontology of Aquela Música was instantiated with data originated from the analysis of a set of songs by Brazilian songwriter Noel Rosa. This choice was due to a few objective criteria, such as: the relevance of this repertoire to Brazilian culture; the fact that the work of aforementioned songwriter is found in public domain, which allows the free use of the songs' lyrics, as well as of the original audio recordings; the diversity and multiplicity of styles and forms existent in this artist's compositions.

Another important aspect behind this choice, although specific of Brazilian culture, is that many of the works by Noel Rosa present an intense trait of social and political chronicle. The representation of such information through the complex category of "themes" made it possible to validate the viability of using the concept of discourse formation to reflect social and cultural aspects. Another important validation was the representation of melodic profiles and of narrative structures, two especially complex categories, for being dependent on one another and also on several other elements of the song, as presented on table 4.

One of the concerns of the construction of the prototype while information system was to make the application of the rules transparent to the final user. By accessing the prototype, the user visualizes the categories and its values indistinctively. With the goal of representing the multidimensional vision of complex thinking, the user can begin browsing from any element or from any song. When attributing value to the desired songs, the system calculates which are the songs that meet more precisely the informed values and presents them prominently. Another concern, in the sense of representing the multidimensional aspect, was to allow the user, from the selection of a determined song, to consult the complete analysis file and combine aspects of the results of these analyses with any other categories of choice.

This way, the user himself simulates the reconstruction path of the songs from the fragments expressed in each of the available elements, which makes it possible to answer the question "What's that song?" formulated in the origin of the project.

4.4 Limitations and Future Works

Due to the fact that it is still a prototype in development, some simplifications were used in the implemented model with respect to the entirety of elements and rules identified in Table 4.

However, the results achieved in the preparation of analyses that take into account the multidisciplinary objects of the songs and the construction of a system capable of expressing the multidimensional nature of a complex reasoning may be considered quite satisfactory and promising.

An important limitation is the method of manual analysis of the songs, which makes it quite laborious to produce content in order to simultaneously feed all dimensions of analysis considered. However, even if applied to a limited repertoire of works by the same songwriter, the results obtained seem auspicious in indicating the feasibility of integrating Aquela Música with other ontology-based information systems, in order to allow the results of the analyzes to be availed and complemented.

Thus, even partial analyzes may be useful when supplemented with the set of information derived from other systems, such as music catalogs or web-based radio applications.

5 Final Considerations

The major challenge on popular music research is mapping the layers of meaning embedded in a musical work, as well as their forms ways of insertion into society and history, avoiding simplifications and analytic mechanisms that may misrepresent their polysemic and complex nature [14].

Therefore, the challenge of constructing popular music classification systems is reflecting the multidimensionality of complex reasoning and encompassing its multiple dimensions.

The research in the Knowledge Organization field, by means of advances in investigations related to the classification systems, points towards considering and integrating the different disciplines involved in the study of the popular song in order to contribute in addressing these challenges by establishing a solid epistemological grounding.

Although it is still a prototype, the Aquele Música system enables to perceive actual perspectives for the building of ontology-based classification systems for complex artistic documents that allow organizing knowledge in a semantic and dynamic way.

Acknowledgements. The Aquela Música project received the “Reflexão Crítica e Produção Cultural para Internet” [Critical Reflection and Cultural Production for the Internet] prize, granted by Fundação Nacional das Artes (Funarte) of the Brazilian Ministry of Culture. Our thanks to Funarte for the recognition and for the prize, which enabled the development of this work.

References

1. Hjørland, B.: Arguments for philosophical realism in library and information science. Library Trends 52, 488–506 (2004)
2. Gomes, H.E.: Tendências da pesquisa em Organização do Conhecimento. Pesquisa Brasileira em Ciência da Informação 2, 60–88 (2009)
3. Lacan, J.: The four fundamental concepts of psycho-analysis. WW Norton & Company (1998)
4. Nestrovski, A.: Lendo música: 10 ensaios sobre 10 canções, pp. 3–7. Publifolha, São Paulo (2007)
5. Morin, E.: Réforme de pensée, transdisciplinarité, réforme de l’Université. In: Congrès International Quelle Université pour demain (1997)
6. Tatit, L.A.M.: Semiótica da Canção: Melodia e Letra. Escuta, São Paulo (1994)
7. Blom, E.: Grove’s Dictionary of Music and Musicians. St. Martin’s Press, New York (1996)
8. Tatit, L.A.M.: Musicando a semiótica. Annablume, São Paulo (1998)

9. Mckay, C., Fujinaga, I.: Musical genre classification: Is it worth pursuing and how can it be improved? In: Proceeding of Seventh International Conference on Music Information Retrieval, pp. 101–106 (2006)
10. Ortiz, R.: A moderna tradição brasileira. Brasiliense, São Paulo (1988)
11. Thompson, A.E.: Playing Tag: An Analysis of Vocabulary Patterns and Relationships Within a Popular Music Folksonomy. University of North Carolina, Chapel Hill (2008)
12. Santini, R.M.: Os usuários e a desorganização da cultura: Os Sistemas de Recomendação e as conseqüências da classificação para os usos sociais da música na Internet. PhD Thesis in Information Science - Universidade Federal Fluminense (Instituto de Arte e Comunicação Social) e Instituto Brasileiro de Informação em Ciência e Tecnologia, Niterói (2010)
13. Fiorin, J.L.: As astúcias da enunciação. Ática, São Paulo (1999)
14. Napolitano, M.: História e Música - história cultural da música popular. Autêntica, Belo Horizonte (2002)
15. Resnick, P.: Recommender Systems. Interview in IOTA. University of Michigan School of Information (1999)
16. Adomavicius, G., Tuzhilin, A.: Toward the next generation of Recommender Systems: A Survey of the State-of-Art and Possible Extension. IEEE Transaction on Knowledge and Data Engeneering 17(6), 734–748 (2005)
17. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: an open architecture for collaborative filtering of Netnews. In: ACM Conference of Computer Supported Cooperative Work, pp. 175–186 (1994)
18. Hill, W., Stead, L., Rosenstein, M., Furnas, G.: Recommending and Evaluating Choices in a Virtual Community of Use. In: Proceedings of Conference Human Factors in Computing Systems (1995)
19. Maes, P., Shardanand, U.: Social Information filtering: algorithms for automating "word of mouth". Human Factors in Computing Systems, 210–217 (1995)
20. LAST.FM. (2013), `http://www.last.fm`
21. Aucouturier, J.-J., Pampalk, E.: From Genres to Tags: a little epistemology of Music Information Retrieval. In: Proceedings of the 9th International Society for Music Information Retrieval Conference, pp. 1–10 (2008)
22. Bosteels, K., Kerre, E., Pampalk, E.: Music Retrieval Based on Social Tags: A Case Study. In: Procceding in ISMIR 2008, Vienna, Austria (2008)
23. LAST.FM. Message from the Last.fm founders - Felix, R. and Martin J. (June 2009), `http://blog.last.fm/`
24. ISMIR. The International Society for Music Information Retrieval (2013), `http://www.ismir.net`
25. Downie, J.S., Byrd, D., Crawford, T.: Ten years of ISMIR: Reflections on challenges and opportunities. In: Proceedings of the 10th International Society for Music Information Retrieval Conference (2009)
26. Foucault, M.: The archaeology of knowledge. Vintage (2012)
27. Tatit, L.A.M.: Analysing popular song. In: Popular Music Studies, pp. 33–50. Arnold, London (2002)
28. Gnoli, C., Bosch, M., Mazzocchi, F.: A new relationship for multidisciplinary knowledge organization systems: dependence. In: Interdisciplinarity and Transdisciplinarity in the Organization of Scientific Knowledge, pp. 399–410 (2007)
29. De Santis, R.: Aquela Música - navegação interativa por canções populares brasileiras. In: Anais do, I. (ed.) Encontro Funarte de políticas para as artes. Funarte, Rio de Janeiro (2011)

30. W3C LDP. Linked Data Platform (LDP) Working Group (2013), http://www.w3.org/2012/ldp/
31. Raimond, Y., Abdallah, S., Sandler, M., Giasson, F.: The music ontology. In: Proceedings of the International Conference on Music Information Retrieval, pp. 417–422 (2007)
32. Brickley, D., Miller, L.: FOAF vocabulary specification 0.98. Namespace Document 9 (2010)
33. Song, S., Minkoo, K., Seungmin, R., Eenjun, H.: Music Ontology for Mood and Situation Reasoning to Support Music Retrieval and Recommendation. In: Third International Conference on Digital Society, pp. 304–309 (2009)
34. Seungmin, R., Byeong-jun, H., Eenjun, H.: SVR-based music mood classification and context-based music recommendation. In: Proceedings of the 17th ACM International Conference on Multimedia, pp. 713–716 (2009)
35. DBTUNE. Music-Related RDF (2013), http://www.dbtune.org

Non-monotonic Reasoning in Conceptual Modeling and Ontology Design: A Proposal

Giovanni Casini[1] and Alessandro Mosca[2]

[1] Centre for Artificial Intelligence Research, CSIR Meraka Institute, South Africa
GCasini@csir.co.za
[2] Free University of Bozen-Bolzano, Faculty of Computer Science, Italy
mosca@inf.unibz.it

Abstract. The *Object Role Modeling language* (ORM2) is nowadays the most widespread fact-based conceptual modeling language in the business world. Recently, it has been proposed an encoding of the core fragment of ORM2 (called ORM2$^{\mathsf{zero}}$) into the description logic $\mathcal{ALCQI}$, allowing the use of reasoning technologies in the analysis of the schemas. A number of services has been defined there based on the FO semantics of ORM2. On the other hand, in many application domains there is a need for the formalization and modeling of *defeasible information* and *non-monotonic* reasoning services. Here we formalize a possible way of introducing non-monotonic reasoning into ORM2 schemas, enriching the language with special set of new constraints.

1 Introduction

ORM2 (Object Role Modelling 2) is a graphical fact-oriented approach for modelling and querying business domain information, which allows for a verbalisation in a controlled natural language easily understandable by non-technical users [9]. The introduction of a formal semantics for ORM2 and the identification of a decidable fragment of it (see [7]) opened the doors for the exploitation of reasoning technologies to support the schema design quality, as well as the possibility to exploit ORM2 as ontology design language. In the last years, especially in the field of ontology design, a lot of attention has been devoted to the implementations of forms of *defeasible reasoning*, and various proposals (*e.g.*, [2,3,5,8]) have been made in order to integrate nonmonotonic reasoning mechanisms into description logics (DLs), the main logic formalism used in ontologies representation.

The paper explores a possible way of implementing non-monotonic reasoning in the ORM2 formalism. Here we focus on the DL-reformulation [5] of the non-monotonic consequence relation called *Rational Closure* (RC) [10], and we show (i) how to model defeasible information in ORM2 and (ii) how to check consistency and draw conclusions from schemas with defeasible information. Due to a lack of space, here we do not enter deeply into the technical details of the proposal, but rather we describe the main idea and its implications in terms of knowledge representation, and we refer to the related literature when needed. The basic idea for the present proposal, together with the defeasible versions

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 361–370, 2013.

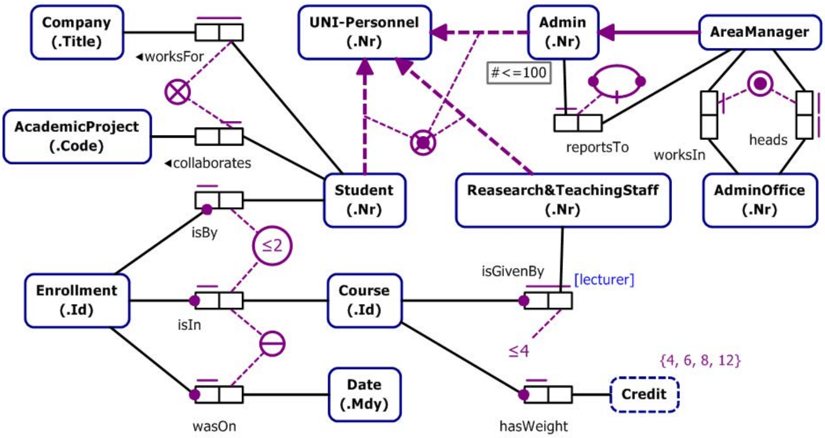


Fig. 1. An ORM2 schema example

of the subtype relation and of the mandatory participation have already been presented in [4].

2 Fact-Oriented Modelling in ORM2

The 'Fact-oriented modelling' is a conceptual modelling approach that views the world in terms of simple facts, *i.e.* assertions about*objects playing* certain *roles* (*e.g.* 'Alice is enrolled in the Computer Science program'). In ORM2, objects may be **entities** (*e.g.* a person or a car) and **values** (*e.g.* a *character string* or a *number*). Moreover, entities and values are described in terms of the **types** they belong to, where a type is a set of instances (entities and values types are called together 'object types'). Each entity in the domain of interest is, therefore, an instance of a particular type. The roles played by the entities in a given domain are introduced by means of **predicates**, where each predicate has a given set of **roles** according to its arity. A role is connected to exactly one object type, indicating that the role is played only by the (possible) instances of that type.

According to the ORM2 design procedure, after the specification of the relevant object types and predicates, *constraints* must be considered. Hereafter, we give an informal introduction of the constraint graphical representation, together with their intended semantics. Fig. 1 shows an example of an ORM2 schema in an 'academic domain' (where the soft rectangles are entity types, the dashed soft rectangles are value types, and the sequences of one or more role-boxes are predicates). The example is not complete w.r.t. the set of all the ORM2 constraints but it aims at giving the feeling of the expressive power of the language. The following are among the constraints included in the schema (see [9] for a comprehensive introduction of the ORM2 constructs, together with their graphical representation):

1. **Subtyping:** solid and dashed arrows representing 'is-a' relationships among types.
2. **Partition:** a combination of an **exclusive** constraint (a circled 'X') saying that 'Research&TeachingStaff, Admin, and Student are *mutually disjoint*', and a **total** constraint (a circled dot) for 'Research&TeachingStaff, Admin, and Student completely *cover* their common super-type'.
3. **Internal frequency occurrence:** *if* an instance of Research'&TeachingStaff plays the role of being lecturer in the relation isGivenBy, that instance can play the role at most 4 times. *At most one* cardinalities (depicted as continuos bars) are special cases of frequency occurrences called **internal uniqueness** constraints.
4. **External frequency occurrence:** attached to the roles played by Student and Course for 'Students are allowed to enrol in the same course *at most* twice'.
5. **External uniqueness:** it is used between the role played by Course in isIn and the role played by Date in wasOn, saying that 'For each combination of Course and Date, *at most one* Enrollment isIn that Course and wasOn that Date'.
6. **Mandatory participation:** a dot saying that 'Each Course is given by *at least one* instance of Research&TeachingStaff'.

3 The $\mathcal{ALCQI}$ Encoding of ORM2^{zero}

With the main aim of relying on effective technologies to reason about ORM2 schemas, an encoding in the DL $\mathcal{ALCQI}$, for which tableaux-based reasoning algorithms with a tractable computational complexity exist, has been devised [7]. $\mathcal{ALCQI}$ corresponds to the basic DL $\mathcal{ALC}$ equipped with *qualified cardinality restrictions* and *inverse roles*, and it is a fragment of the OWL2 web ontology language (a complete introduction of the syntax and semantics of $\mathcal{ALCQI}$ can be found in [1]). Apart from the fact that the full ORM2 language is undecidable for several reasons [7], the difference in expressivity between $\mathcal{ALCQI}$ and ORM2 is substantial: $\mathcal{ALCQI}$ does not admit neither set-comparison, nor frequency occurrence statements, about arbitrary role sequences. And moreover, n-ary relations are not part of the $\mathcal{ALCQI}$ language, and reified relations must be introduced in the encoding[1]. Due to this limitations, the fragment called ORM2^{zero}, which is maximal with respect to the expressiveness of $\mathcal{ALCQI}$ and still expressive enough to capture the most frequent usage patterns of the conceptual modelling community, has been identified.

ORM2^{zero} corresponds to the fragment of ORM2 equipped with typing, mandatory, subtyping (i.e., exclusive, exhaustive, and partition) constraints, and a restricted version of the set-comparison (i.e., subset, equality, and exclusion) and of the frequency occurrence constraints. The encoding of the semantics of ORM2^{zero} is shown in Table 3 where: (i) $E_1, \ldots, E_n$ are *entity type* concepts; (ii) $V_1, \ldots, V_m$ are *value type* concepts; (iii) $A_{R_1}, \ldots, A_{R_k}$ are reified n-ary *relations*; (iv) $D_1, D_2, \ldots, D_h$ are *domain symbol* concepts; and (v) $1, \ldots, n_{max} + 1$ are DL roles. There, additional *background axioms* are needed in order to force the interpretation of the $\mathcal{ALCQI}$ KB to be correct w.r.t. the corresponding ORM2^{zero} schema (the complete proof of the correctness theorem is available at [6]).

[1] I.e., for each relation R of arity $n \geq 2$, a new atomic concept A_R and n functional roles are introduced.

Table 1. $\mathcal{ALCQI}$ encoding

Background axioms:	$E_i \sqsubseteq \neg(D_1 \sqcup \cdots \sqcup D_l)$ for $i \in \{1, \ldots, n\}$ $V_i \sqsubseteq D_j$ for $i \in \{1, \ldots, m\}$, and some j with $1 \leq j \leq l$ $D_i \sqsubseteq \sqcap_{j=i+1}^{l} \neg D_j$ for $i \in \{1, \ldots, l\}$ $\top \sqsubseteq A_{\top_1} \sqcup \cdots \sqcup A_{\top_{n_{max}}}$ $\top \sqsubseteq (\leq 1 i.\top)$ for $i \in \{1, \ldots, n_{max}\}$ $\forall i.\bot \sqsubseteq \forall i+1.\bot$ for $i \in \{1, \ldots, n_{max}\}$ $A_{\top_n} \equiv \exists 1.A_{\top_1} \sqcap \cdots \sqcap \exists n.A_{\top_1} \sqcap \forall n+1.\bot$ for $n \in \{2, \ldots, n_{max}\}$ $A_R \sqsubseteq A_{\top_n}$ for each atomic relation R of arity n $A \sqsubseteq A_{\top_1}$ for each atomic concept A	
TYPE$(R.a, O)$	$\exists \tau(R.a)^-.A_R \sqsubseteq O$	
FREQ$^-(R.a, \langle \text{min}, \text{max} \rangle)$	$\exists \tau(R.a)^-.A_R \sqsubseteq\ \geq \text{min}\ \tau(R.a)^-.A_R \sqcap \leq \text{max}\ \tau(R.a)^-.A_R$	
MAND$(\{R^1.a_1, \ldots, R^1.a_n,$ $\ldots, R^k.a_1, \ldots, R^k.a_m\}, O)$	$O \sqsubseteq \exists \tau(R^1.a_1)^-.A_{R^1} \sqcup \cdots \sqcup \exists \tau(R^1.a_n)^-.A_{R^1} \sqcup \cdots \sqcup$ $\exists \tau(R^k.a_1)^-.A_{R^k} \sqcup \cdots \sqcup \exists \tau(R^k.a_m)^-.A_{R^k}$	
R-SET$^-_{\text{Sub}}(A, B)$ R-SET$^-_{\text{Exc}}(A, B)$	$A_R \sqsubseteq A_S$ $A_R \sqsubseteq A_{\top_n} \sqcap \neg A_S$	(entire relations)
R-SET$^-_{\text{Sub}}(A, B)$ R-SET$^-_{\text{Exc}}(A, B)$	$\exists \tau(R.a_i)^-.A_R \sqsubseteq \exists \tau(S.b_j)^-.A_S$ $\exists \tau(R.a_i)^-.A_R \sqsubseteq A_{\top_n} \sqcap \neg \exists \tau(S.b_j)^-.A_S$	(pair of roles)
O-SET$_{\text{Isa}}(\{O_1, \ldots, O_n\}, O)$	$O_1 \sqcup \cdots \sqcup O_n \sqsubseteq O$	
O-SET$_{\text{Tot}}(\{O_1, \ldots, O_n\}, O)$	$O \sqsubseteq O_1 \sqcup \cdots \sqcup O_n$	
O-SET$_{\text{Ex}}(\{O_1, \ldots, O_n\}, O)$	$O_1 \sqcup \cdots \sqcup O_n \sqsubseteq O$ and $O_i \sqsubseteq \sqcap_{j=i+1}^{n} \neg O_j$ for each $i = 1, \ldots, n$	
OBJ(R, O)	$O \equiv A_R$	

It is finally important to say that in ORM2, and in conceptual modeling languages in general, the notion of consistency is slightly different from the classical logical one. As a matter of fact, from a logical point of view a KB $\mathcal{K}$ is considered inconsistent only if we can classically derive a contradiction from it ($\mathcal{K} \models \top \sqsubseteq \bot$ in DL). Instead, dealing with conceptual schemas we generally desire that they satisfy a stronger form of consistency, that is, we want that none of the classes in the schema is forced to be empty.

Definition 1 (Strong consistency). *A TBox $\mathcal{T}$ is* strongly consistent *if none of the atomic concepts present in its axioms are forced to be empty, that is, if $\mathcal{T} \not\models A \sqsubseteq \bot$ for every atomic concept A appearing in the inclusion axioms in $\mathcal{T}$.*

4 Rational Closure in $\mathcal{ALCQI}$

It's time now to briefly present the specialization of the RC [10] procedure for the DL $\mathcal{ALCQI}$. Among the various proposals in non-monotonic reasoning, we have chosen RC because of its interesting characteristics: (i) the related consequence relation satisfies important logical properties [10,5]; (ii) the conclusions one can draw are intuitive; (iii) the procedure can be reduced to a series of decisions w.r.t. the classical $\mathcal{ALCQI}$ consequence relation $\models$.

A TBox $\mathcal{T}$ for $\mathcal{ALCQI}$ consists of a finite set of general inclusion axioms (GCIs) of form $C \sqsubseteq D$ (C and D being concepts). Now we introduce also the *defeasible inclusion axiom* $C \precsim D$, that is read as 'Typically, an individual falling under the concept C falls also under the concept D'. We indicate with $\mathcal{D}$ (DBox) the finite set of such inclusion axioms.

Example 1. Consider a modification of the classical 'penguin example', with the concepts P, B, F, I, Fi, respectively read as 'penguin', 'bird', 'flying', 'insect', and 'fish', and a role $Prey$, where a role instantiation $(a,b){:}Prey$ read as 'a preys for b'. We can define a defeasible knowledge base (KB) $\mathcal{K} = \langle \mathcal{T}, \mathcal{D} \rangle$ with $\mathcal{T} = \{P \sqsubseteq B, I \sqsubseteq \neg Fi\}$ and $\mathcal{D} = \{P \precsim \neg F,\ B \precsim F,\ P \precsim \forall Prey.Fi \sqcap \exists Prey.\top, B \precsim \forall Prey.I \sqcap \exists Prey.\top\}$.

RC is a logical procedure that, given a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{D} \rangle$, decides if a defeasible axiom $C \precsim D$ is derivable or not from the KB. The aim is to resolve potential conflicts in our KB (*e.g.* penguins do not fly, but, being a subtype of birds, we could also derive that they fly). The basic idea is to order the defeasible information w.r.t. its specificity: an *exceptionality ranking* of the axioms in $\mathcal{D}$ starting from the most general ones (*e.g.* the ones that give us information about birds) up to more specific ones (*e.g.* the ones that give us information about penguins). In case of potential conflicts (*e.g.*, the conclusion that penguins fly and do not fly), the procedure retains the more specific defeasible information (penguins typically do not fly), eliminating the more general one (birds typically fly). Due to the limits in space, we present here a minimal technical explanation of the procedure, referring the reader to [5] for a more extensive presentation of the technicalities and the intuitions behind the procedure.

Given a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{D} \rangle$ and a query $C \precsim D$, the procedure can be divided into two phases: (i) the definition of the *exceptionality ranking* of the axioms in $\mathcal{D}$; (ii) the decision whether $C \precsim D$ is or is not in the RC of $\mathcal{K}$.

(i) A concept is considered *exceptional* in a KB $\langle \mathcal{T}, \mathcal{D} \rangle$ if it is classically negated, that is, C is exceptional in $\langle \mathcal{T}, \mathcal{D} \rangle$ if

$$\models \bigsqcap \overline{\mathcal{T}} \sqcap \bigsqcap \overline{\mathcal{D}} \sqcap C \sqsubseteq \bot$$

where $\overline{\mathcal{T}} = \{\neg C \sqcup D \mid C \sqsubseteq D \in \mathcal{D}\}$, $\overline{\mathcal{D}} = \{\neg C \sqcup D \mid C \precsim D \in \mathcal{D}\}$, and $\models$ is the classical consequence relation associated to $\mathcal{ALCQI}$. In turn, a defeasible axiom is considered exceptional if its antecedent is exceptional. Let E be a function that, given $\langle \mathcal{T}, \mathcal{D} \rangle$ gives back the exceptional axioms in $\mathcal{D}$ ($E(\mathcal{D}) = \{C \precsim D \mid C$ exceptional in $\langle \mathcal{T}, \mathcal{D} \rangle)$. We can define a sequence $\mathcal{E}_0, \mathcal{E}_1, \ldots$ of subsets of $\mathcal{D}$ s.t. $\mathcal{E}_0 = \mathcal{D}$ and $\mathcal{E}_{i+1} = E(\mathcal{E}_i)$. Since $\mathcal{D}$ is a finite set, the procedure terminates with a (possibly empty) fixed point of E, that we call $\mathcal{E}_\infty$.

(ii) Given a query $C \precsim D$, we check at which level of exceptionality we have to position the antecedent C, that is, we associate C with the defeasible information contained in the lowest $\mathcal{E}_i$ s.t.:

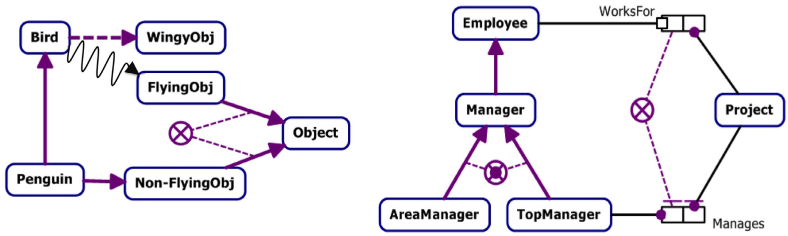
$$\not\models \bigsqcap \overline{\mathcal{T}} \sqcap \bigsqcap \overline{\mathcal{E}_i} \sqcap C \sqsubseteq \bot$$

Hence, we can decide if the defeasible inclusion axiom $C \precsim D$ is in the rational closure of $\langle \mathcal{T}, \mathcal{D} \rangle$:

$$\langle \mathcal{T}, \mathcal{D} \rangle \Vdash_r C \precsim D \textit{ iff } \ \models \bigsqcap \overline{\mathcal{T}} \sqcap \bigsqcap \overline{\mathcal{E}_i} \sqcap C \sqsubseteq D$$

where $\Vdash_r$ indicates the inference relation characterizing the Rational Closure.

Example 2. Consider the KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{D} \rangle$ in Example 1. $\overline{\mathcal{T}} = \{\neg P \sqcup B, \neg I \sqcup \neg Fi\}$ and $\overline{\mathcal{D}} = \{\neg P \sqcup \neg F,\ \neg B \sqcup F,\ \neg P \sqcup (\forall Prey.Fi \sqcap \exists Prey.\top), \neg B \sqcup (\forall Prey.I \sqcap \exists Prey.\top)\}$.



(a) (b)

Fig. 2. (a) Defeasible subtyping (b) Defeasible mandatory

Table 2. $\mathcal{ALCQI}$ encoding of the $\mathsf{ORM2}^{\mathsf{zero}}$ 'Non-Flying Birds' example

Subtyping: $\mathsf{FlyingObject} \sqcup {}^{\mathsf{Non}}\mathsf{FlyingObject} \sqsubseteq \mathsf{Object}$
$\mathsf{FlyingObject} \sqsubseteq \neg {}^{\mathsf{Non}}\mathsf{FlyingObject}$
$\mathsf{Penguin} \sqsubseteq \mathsf{Bird}$, $\mathsf{Penguin} \sqsubseteq {}^{\mathsf{Non}}\mathsf{FlyingObject}$
$\mathsf{Bird} \sqsubseteq \mathsf{WingyObject}$, $\mathsf{Bird} \sqsubseteq \mathsf{FlyingObject}$

We obtain the exceptionality ranking of the sequents: $\mathcal{E}_0 = \{B \tilde{\sqsubseteq} F, B \tilde{\sqsubseteq} \forall Prey.I \sqcap \exists Prey.\top, P \tilde{\sqsubseteq} \neg F, P \tilde{\sqsubseteq} \forall Prey.Fi \sqcap \exists Prey.\top\}$; $\mathcal{E}_1 = \{P \tilde{\sqsubseteq} \neg F, P \tilde{\sqsubseteq} \forall Prey.Fi \sqcap \exists Prey.\top\}$.

Assume we want to check the properties of penguins (concept P). Hence, we have to find the exceptionality level of P, that is 1 since $\models \bigsqcap \overline{\mathcal{T}} \sqcap \bigsqcap \overline{\mathcal{E}_0} \sqcap P \sqsubseteq \bot$, but $\not\models \bigsqcap \overline{\mathcal{T}} \sqcap \bigsqcap \overline{\mathcal{E}_1} \sqcap P \sqsubseteq \bot$. Associating to P the defeasible information in $\mathcal{E}_1$ we are able to derive that penguins do not fly and eat fishes, but not that penguins fly and eat insects.

5 Defeasible Constraints for ORM2

This section introduces a new set of ORM2 defeasible constraints. Such constraints have exactly the same meaning of the defeasible inclusion axioms defined above: the constraint *typically* holds, but there could be exceptional cases that do not respect it.

Defeasible subtyping relation (a wavy arrow instead of the standard one).The wavy arrow indicates that each element of the class C is also an element of the class D, if not informed of the contrary. While the classical subtype relation is encoded as $C \sqsubseteq D$, the new defeasible connection is encoded by $C \tilde{\sqsubseteq} D$.

Example 3 (Defeasible subtype relation). Consider Fig. 2(a). The ORM2 schema represents the classic penguin example: penguins are birds and do not fly, while birds fly and have wings. The encoding procedure of the classical ORM2 schema into $\mathcal{ALCQI}$ gives back the TBox $\mathcal{T}$ in Table 2, that implies $\mathcal{T} \models \mathsf{Penguin} \sqsubseteq \bot$, *i.e.* the concept Penguin must be empty. We can modify the KB introducing defeasible information, in particular stating that birds *typically* fly. In such a way we obtain a new KB with $\mathsf{Bird} \tilde{\sqsubseteq} \mathsf{FlyingObject}$, substituting the corresponding classical axiom. Now we can derive the same kind of conclusions as in the example 2 (*e.g.*, we cannot derive that penguins fly).

Table 3. $\mathcal{ALCQI}$ encoding of the 'Non-Managing Employees' example

Typing:	$\mathsf{WorksFor} \sqsubseteq \exists \mathsf{f1}^{-}.\mathsf{Employee}$, $\mathsf{WorksFor} \sqsubseteq \exists \mathsf{f2}^{-}.\mathsf{Project}$ $\mathsf{Manages} \sqsubseteq \exists \mathsf{f1}^{-}.\mathsf{TopManager}$, $\mathsf{Manages} \sqsubseteq \exists \mathsf{f2}^{-}.\mathsf{Project}$
Frequency:	$\exists \mathsf{f1}^{-}.\mathsf{Manages} \sqsubseteq\ = 1\, \mathsf{f1}^{-}.\mathsf{Manages}$
Mandatory:	$\mathsf{Employee} \sqsubseteq \exists \mathsf{f1}^{-}.\mathsf{WorksFor}$ $\mathsf{TopManager} \sqsubseteq \exists \mathsf{f1}^{-}.\mathsf{Manages}$ $\mathsf{Project} \sqsubseteq \exists \mathsf{f2}^{-}.\mathsf{WorksFor}$ $\mathsf{Project} \sqsubseteq \exists \mathsf{f2}^{-}.\mathsf{Manages}$
Exclusion:	$\exists \mathsf{f1}^{-}.\mathsf{WorksFor} \sqsubseteq \mathsf{A}_{\top 2} \sqcap \neg \exists \mathsf{f1}.\mathsf{Manages}$
Subtyping:	$\mathsf{Manager} \sqsubseteq \mathsf{Employee} \sqcap (\mathsf{AreaManager} \sqcup \mathsf{TopManager})$ $\mathsf{AreaManager} \sqsubseteq \neg \mathsf{TopManager}$

Defeasible Mandatory Participation. We introduce a defeasible version (□) of the mandatory participation constraint (•). If the connection between a class C and a relation R is accompanied by the defeasible version of the mandatory constraint, it is read as 'each element of the class C participates to the relation R, if we are not informed of the contrary'. The mandatory participation of the class B to the role $\mathsf{A_N}$ is encoded into the axiom $\mathsf{B} \sqsubseteq \exists \mathsf{f1}^{-}.\mathsf{A_N}$, hence we make it defeasible using $\mathsf{B} \mathrel{\tilde{\sqsubseteq}} \exists \mathsf{f1}^{-}.\mathsf{A_N}$.

Example 4 (Defeasible mandatory participation). Consider Fig. 2(b). The schema represents the organization of a firm: the class Manager is a subtype of the class Employee, and every employee *must* work for a project. On the other hand, every top manager mandatorily manages a project, and TopManager is a subtype of Manager. The correspondent $\mathcal{ALCQI}$ TBox $\mathcal{T}$ is in table 3. Since managing and working for a project are not compatible roles, $\mathcal{T} \models \mathsf{TopManager} \sqsubseteq \bot$. Instead, if we declare that *typically* an employee works for a project, we end up considering the top managers as exceptional employees, and we obtain a KB as the one in Table 3, but with $\mathsf{Employee} \mathrel{\tilde{\sqsubseteq}} \exists \mathsf{f1}^{-}.\mathsf{WorksFor}$ instead of $\mathsf{Employee} \sqsubseteq \exists \mathsf{f1}^{-}.\mathsf{WorksFor}$. Since TopManager is not consistent with the TBox plus the defeasible axiom, we cannot associate the latter to TopManager and, despite we know that normally an employee works for a project, we are not forced to such a conclusion about the top managers.

Defeasible Internal Frequency Occurrence (a wavy line instead of the dashed one). We can introduce a form of defeasible frequency constraint, that binds the number of times an instance of a particular type can play a certain role, if we have not more specific information imposing different constraints. The frequency constraint stating how many times each instance of a type B can play a role $\mathsf{A_N}$ is encoded by $\exists \mathsf{f1}^{-}.\mathsf{A_N} \sqsubseteq = \mathsf{i}\ \mathsf{f1}^{-}.\mathsf{A_N}$ ($i \in \mathbb{N}$), and hence made defeasible using $\exists \mathsf{f1}^{-}.\mathsf{A_N} \mathrel{\tilde{\sqsubseteq}}\ = \mathsf{i}\ \mathsf{f1}^{-}.\mathsf{A_N}$.

Example 5 (Defeasible internal frequency occurrence). A taxonomization problem about bee species in sub-Saharan Africa: most male afrotropical bees have thirteen segment antennae (SegAnt), but male belonging to the pasite genus (PGBee), that belongs to the apidae family (ApiFBee) of the afrotropical super family, has only twelve segment antennae. We can formalize such information in the ORM2 schema in Fig. 3(a),

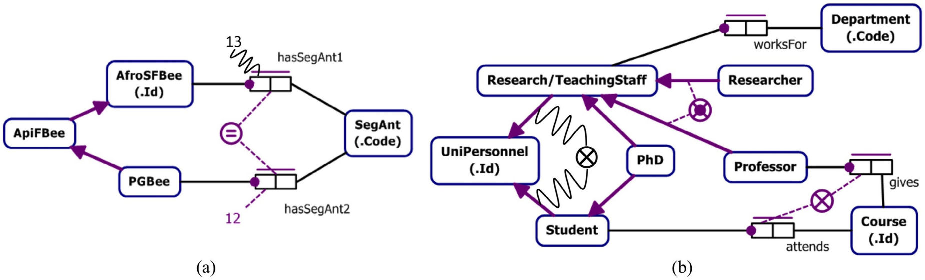


Fig. 3. (a) Defeasible frequency occurrence (b) Defeasible exclusive subtyping

Table 4. $\mathcal{ALCQI}$ encoding of the 'Bees classification' example

Typing:	$\mathsf{HasSegAnt1} \sqsubseteq \exists \mathsf{f1}^-.\mathsf{AfroSFBee}$, $\mathsf{HasSegAnt1} \sqsubseteq \exists \mathsf{f2}^-.\mathsf{SegAnt}$ $\mathsf{HasSegAnt2} \sqsubseteq \exists \mathsf{f1}^-.\mathsf{PGBee}$, $\mathsf{HasSegAnt2} \sqsubseteq \exists \mathsf{f2}^-.\mathsf{SegAnt}$
Frequency:	$\exists \mathsf{f1}^-.\mathsf{HasSegAnt1} \sqsubseteq\, = 13\, \mathsf{f1}^-.\mathsf{HasSegAnt1}$ $\exists \mathsf{f1}^-.\mathsf{HasSegAnt2} \sqsubseteq\, = 12\, \mathsf{f1}^-.\mathsf{HasSegAnt2}$
Mandatory:	$\mathsf{AfroSFBee} \sqsubseteq \exists \mathsf{f1}^-.\mathsf{HasSegAnt1}$ $\mathsf{PGBee} \sqsubseteq \exists \mathsf{f1}^-.\mathsf{HasSegAnt2}$
Equality:	$\mathsf{HasSegAnt1} \equiv \mathsf{HasSegAnt2}$
Subtyping:	$\mathsf{PGBee} \sqsubseteq \mathsf{ApiFBee}$, $\mathsf{ApiFBee} \sqsubseteq \mathsf{AfroSFBee}$

but such a schema is not strongly consistent, since its translation in $\mathcal{ALCQI}$ (Table 4) implies $\mathsf{PGBee} \sqsubseteq \bot$. Introducing a form of defeasible frequency occurrence for the more general type $\mathsf{AfroSFBee}$, that is encoded into the defeasible inclusion axiom $\exists \mathsf{f1}^-.\mathsf{HasSegAnt1} \mathrel{\tilde{\sqsubseteq}}\, = 13\ \mathsf{f1}^-.\mathsf{HasSegAnt1}$ we preserve strong consistency, since the type PGBee turns out to be exceptional, and we don't have to consider the defeasible axiom when reasoning about PGBee. Hence we can derive that, if we are not informed of the contrary, male afrotropical bees have thirteen segment antennae, while the males of the pasite genus have twelve segment antennae, but still being a subtype of $\mathsf{AfroSFBee}$.

Defeasible Disjointness (wavy lines instead of the dashed ones). It's a defeasible exclusive constraint, stating that two types or the participation to two distinct roles are mutually disjoint, if not informed of the contrary. In the $\mathcal{ALCQI}$ encoding, an axiom $A \sqsubseteq \neg B$ is changed into an axiom $A \mathrel{\tilde{\sqsubseteq}} \neg B$

Example 6 (Defeasible disjointness). In the university we distinguish between the two disjoint classes of students and research/teaching staff, in turn partitioned into professors and researchers. Assume we add to the schema the class of the PhD students, that are students, but are also part of the research/teaching staff. In the classical formulation we would derive $\mathsf{PhD} \sqsubseteq \bot$, since students and research staff must be disjoint class, but if we make defeasible the disjointness axiom ($\mathsf{Student} \mathrel{\tilde{\sqsubseteq}} \neg \mathsf{R\&TStaff}$), we conclude that normally the individuals in one class cannot be in the other, but the PhD students are an exceptional class that shares the properties of both the students and

Table 5. $\mathcal{ALCQI}$ encoding of the 'University Personnel' example

Typing:	$\mathsf{WorksFor} \sqsubseteq \exists \mathsf{f1}^-.\mathsf{R\&TStaff}$, $\mathsf{WorksFor} \sqsubseteq \exists \mathsf{f2}^-.\mathsf{Department}$ $\mathsf{Attends} \sqsubseteq \exists \mathsf{f1}^-.\mathsf{Student}$, $\mathsf{Attends} \sqsubseteq \exists \mathsf{f2}^-.\mathsf{Course}$ $\mathsf{Gives} \sqsubseteq \exists \mathsf{f1}^-.\mathsf{Professor}$, $\mathsf{Gives} \sqsubseteq \exists \mathsf{f2}^-.\mathsf{Course}$
Mandatory:	$\mathsf{R\&TStaff} \sqsubseteq \exists \mathsf{f1}^-.\mathsf{WorksFor}$ $\mathsf{Professor} \sqsubseteq \exists \mathsf{f1}^-.\mathsf{Gives}$ $\mathsf{Student} \sqsubseteq \exists \mathsf{f1}^-.\mathsf{Attends}$
Exclusion:	$\exists \mathsf{f1}^-.\mathsf{Attends} \sqsubseteq \mathsf{A}_{\top 2} \sqcap \neg \exists \mathsf{f1}^-.\mathsf{Professor}$
Subtyping:	$\mathsf{Researcher} \sqsubseteq \mathsf{R\&TStaff} \sqcap \neg \mathsf{Professor}$, $\mathsf{Professor} \sqsubseteq \mathsf{R\&TStaff}$ $\mathsf{R\&TStaff} \sqsubseteq \mathsf{UniPersonnel} \sqcap (\mathsf{Researcher} \sqcup \mathsf{Professor})$ $\mathsf{Student} \sqsubseteq \mathsf{UniPersonnel}$, $\mathsf{Student} \sqsubseteq \neg \mathsf{R\&TStaff}$

the research/teaching staff. Moreover, since the research/teaching staff is partitioned into professors and researches, and since giving courses, a mandatory property of the professors, is not compatible with attending to them, we can derive that PhD students are a subtype of the researchers.

Consistency. As seen at the end of Sec. 3, the notion of consistency we have to deal with is the *strong consistency*, that is, a schema is consistent if we are not forced to conclude about any type that it is empty. The examples above show that the introduction of defeasible constraints into $\mathsf{ORM2}^{\mathsf{zero}}$ allows to build schemas that in the standard notation would be considered inconsistent (w.r.t. strong consistency), but that, once introduced the defeasible constraints, allow for an instantiation such that all the classes can be non-empty. Hence it is necessary to redefine the notion of consistency check in order to deal with such situations. If one decide to rely on the ranking procedure presented above, it is sufficient to check the exceptionality ranking of the KB: if a concept C is exceptional, then it represents an atypical situation, but that is compatible with the information conveyed by the defeasible inclusion axioms (consider the exceptional types in our examples). In RC the only case in which a concept appearing in the axioms is necessarily empty is when $\mathcal{E}_\infty \neq \emptyset$, *i.e.*, despite we eliminate all the defeasible axioms we are allowed to, the antecedents of the axioms in $\mathcal{E}_\infty$ still are negated, *i.e.*, they are empty even in the most exceptional situations. Note that schema that it's inconsistent w.r.t. its non-defeasible part, is inconsistent also in the following new definition.

Definition 2 (Strong consistency). *A KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{D} \rangle$ is* strongly consistent *if none of the atomic concepts present in its axioms are forced to be empty, that is, if $\mathcal{E}_\infty \neq \emptyset$.*

6 Conclusions and Further Work

We have presented a way to implement a form of defeasible reasoning into the $\mathsf{ORM2}$ formalism. Exploiting the possibility of encoding $\mathsf{ORM2}^{\mathsf{zero}}$ into the description logic $\mathcal{ALCQI}$ on one hand, and a procedure appropriate for modeling the RC into DLs on the other, we have defined a set of new $\mathsf{ORM2}$ constraints that are appropriate for modeling defeasible information. Once translated into $\mathcal{ALCQI}$, these constraints allow to use the procedures characterizing the RC to reason on ORM2$^{\text{zero}}$ schemas.

The present proposal deals only with reasoning on the information contained in the TBox obtained from an ORM2 schema, but, once we have done the RC of the TBox, we can think also of introducing an ABox, that is, the information about a particular domain of individuals; a first proposal in such direction is in [5].

An immediate extension of the present work will be to determine if there are others ORM2 constraints that could be introduced also in their defeasible versions: we have to consider such constraints both from the modeling and the technical point of view (*i.e.*, we have to check if such defeasible constraints make sense from the perspective of the modeler and if they are implementable in our logical framework). Since all the constraints in an ORM2$^{\text{zero}}$ schema are encoded into inclusion axioms $C \sqsubseteq D$, in principle every ORM2$^{\text{zero}}$ constraint can be modified into a defeasible version using $\mathrel{\underset{\sim}{\sqsubset}}$, but we need feedback from the ORM2 community in order to understand which defeasible constraints would be desirable.

The introduction of defeasible constraints in other conceptual modeling languages as ER and UML is another possible development. Eventually, it would be interesting to investigate the use of defeasible constraints as a revision tool: dealing with strong consistency, in many cases we have the possibility to transform an inconsistent schema into a consistent one just substituting some classical constraints with their defeasible versions, instead of simply eliminating them.

References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The description logic handbook. Cambridge University Press, USA (2003)
2. Bonatti, P.A., Faella, M., Sauro, L.: Defeasible inclusions in low-complexity DLs. JAIR 42, 719–764 (2011)
3. Britz, K., Meyer, T., Varzinczak, I.: Semantic foundation for preferential description logics. In: Wang, D., Reynolds, M. (eds.) AI 2011. LNCS, vol. 7106, pp. 491–500. Springer, Heidelberg (2011)
4. Casini, G., Mosca, A.: Defeasible reasoning in ORM2. In: Proceedings of AOW 2012, pp. 4–15. CEUR (2012)
5. Casini, G., Straccia, U.: Rational closure for defeasible description logics. In: Janhunen, T., Niemelä, I. (eds.) JELIA 2010. LNCS, vol. 6341, pp. 77–90. Springer, Heidelberg (2010)
6. Franconi, E., Mosca, A.: The formalisation of ORM2 and its encoding in OWL2. Technical Report 12-2, FUB (2012), `http://www.inf.unibz.it/krdb/pub/TR/KRDB12-2.pdf`
7. Franconi, E., Mosca, A., Solomakhin, D.: ORM2: formalisation and encoding in OWL2. In: Herrero, P., Panetto, H., Meersman, R., Dillon, T. (eds.) OTM 2012. LNCS, vol. 7567, pp. 368–378. Springer, Heidelberg (2012)
8. Giordano, L., Olivetti, N., Gliozzi, V., Pozzato, G.L.: ALC + T: a preferential extension of description logics. Fundam. Inform. 96(3), 341–372 (2009)
9. Halpin, T., Morgan, T.: Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design, 2nd edn. Morgan Kaufmann (2001)
10. Lehmann, D., Magidor, M.: What does a conditional knowledge base entail? Artif. Intell. 55(1), 1–60 (1992)

Process Variability through Automated Late Selection of Fragments

Aitor Murguzur[1], Goiuria Sagardui[2], Karmele Intxausti[1], and Salvador Trujillo[1]

[1] IK4-IKERLAN Research Centre, Arrasate-Mondragón, Spain
{amurguzur,kintxausti,strujillo}@ikerlan.es
[2] Mondragon University, Arrasate-Mondragón, Spain
gsagardui@mondragon.edu

Abstract. Process-aware information systems must encompass business process flexibility support due to business needs and factors coming from assorted sources, changing market conditions, customer needs, and regulations. However, flexibility may not be always achieved by pre-specified processes whereby, when context information is only available at runtime, decision making should be deferred to execution time. The late selection pattern defers the selection of placeholder activities' implementations, binding applicable process fragments at runtime. This paper presents the foundations of a novel approach for an end-to-end variability management of process models through late selection of fragments by means of: (i) managing process fragments separately from the base model, (ii) resolving variation points automatically considering constraints and context data at runtime, and (iii) enabling process fragment recommendations based on experience logs.

Keywords: Business Process Flexibility, Late Binding, Runtime Variability, Process Mining.

1 Introduction

Nowadays, organizations have to deal with constant change and uncertainty so as to adapt and gain competitive advantage to ensure their future, i.e., they need to deal with the increasing challenges of customer satisfaction, market evolution, and new requirements. At the same time, they are forced to deal with a growing rate of change in which they need to react to new market circumstances. In those dynamic business environments, Process-aware Information Systems (PAISs) should provide flexibility [1].

The notion of flexibility has emerged as a main research topic in Business Process Management (BPM) over the last decade [2]. Many research approaches for business process (BP) flexibility have been proposed and some of them have been implemented in commercial [3] and open-source community [4,5,6] flexible PAISs. However, in dynamic business environments where neither processes can be fully pre-specified nor process variants can be established, fully structured

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 371–385, 2013.

processes are too restrictive and have problems dealing with change [7]. Therefore, PAISs must be able to deal with uncertainty, deferring decisions regarding the exact control flow to runtime and providing a certain degree of looseness. This looseness may be achieved by deferring placeholder activities binding (Late Selection, also referred to as *runtime variability*), completing the composition at runtime (Late Modeling), or allowing the whole process creation at execution phase (Late Composition) [8].

Concerning those flexibility patterns, there exist trade-offs regarding degree of control, expressiveness and user support, which involve the selection of a particular pattern for defining and executing loosely-specified processes in dynamic business environments [1]. In Late Modeling/Composition patterns compared to Late Selection, the degree of decision deferral is further increased. Hence, instead of just being able to select from a set of pre-specified process fragments or activity implementations, respective fragments may be modeled or composed dynamically during enactment. The largest degree of freedom is provided by Late Composition which can be useful in highly dynamic business environments, due to lack of domain knowledge or given the complexity of task combination. Nevertheless, composition task gives rise to a number of drawbacks. The computation cost may be unacceptable when the number of tasks is fairly large, which directly impacts scalability. In Late Selection of process fragments, the number of variants for each placeholder activity is narrow, so scalability problems may be reduced and system's performance may be better as composition needs higher computational memory. Moreover, although the planner gets most of the time a solution during the composition task, the solution may not be always feasible (known NP-hard problem).

Taking Late Selection pattern as a base, this paper introduces the LateVa approach (**Late Va**riability of Process Models through Automated Selection of Fragments). In particular, it provides an automated solution for selecting the most appropriate implementation from a set of available fragments. LateVa approach also allows variability modeling of business processes by describing all possible variation points and applicable fragments.

The remainder of the paper is organized as follows. In Section 2 we present the background and motivation of this work. Then, we use an elevator maintenance case study in Section 3 to exemplify our findings. After, in Section 4, we introduce the foundations of the LateVa approach, describing loosely-specified process models phases: modeling, execution, and evaluation. Section 5 discusses related work, and finally, in Section 6, the paper draws conclusions and summarizes future plans.

2 Background and Motivation

2.1 Flexibility in PAISs

Over the last several years BP flexibility has attracted significant attention which has resulted in the emergence of a number of taxonomies [9,8]. The taxonomy of Reichert et al. [8] basically differs four types of process flexibility:

1. **Variability:** Variability handles different process variants depending on a particular context [10]. Process variants essentially share a common part of a core process whereas, concrete parts fluctuate from variant to variant, using a *behavioral* (i.e., defining all configurations in a single process model) or *structural* approach (i.e., using different models for variability representation).
2. **Adaptation:** Adaptation represents the ability to deal with changes and consequently adapt process behavior and its structure. Two types of drivers trigger process adaptation: (i) *special situations*, which are not foreseen in the process model, and (ii) *exceptions*, which are foreseen and often require process deviations. Those exceptions may be anticipated or not, however in most of the cases it is hardly to foresee all exceptions, so unanticipated exceptions support is required.
3. **Evolution:** Evolution represents the ability of process instances to change when the corresponding process schema evolves. Business processes can evolve over time, so it is necessary to include mechanisms to support evolutionary changes. In this evolution case, the assumption is that process model changes will affect new and running instances, migrating later instances to the new specification.
4. **Looseness:** Looseness is related to knowledge-intensive processes which support loosely-specified BP models. Different decision deferral patterns might be used to achieve loosely-specified models: (i) *Late Selection* deferring the selection of a placeholder activity to runtime, (ii) *Late Modeling* deferring the modeling of a placeholder activity to runtime, and (iii) *Late Composition* deferring the whole plan creation to runtime.

Traditional PAISs have focused on repetitive and predictable processes [1]. Still, such processes need to operate in dynamic business environments where flexibility becomes a prerequisite, healthcare offers a clear example [11]. For instance, in the diagnostic and treatment process of a particular patient where various organizationally somewhat separate units are involved, a pre-specified process for a case treatment can often be influenced by tests, drug's side effects or patient's progress. Consequently, respective BP flexibility needs should not be ignored with the aim of offering an elastic solution able to adapt to changes and cope with uncertainty in such dynamic conditions.

2.2 Dynamic Business Environments

In our view, featuring the business changing environments, the dynamic nature results from three main factors (see Fig. 1):

– **Degree of change of the environment:** The degree of change of the environment is related to the number of changes that happen in a specific domain.
– **Degree of change predictability:** The degree of change predictability refers to how likely is to find out in advance changes that occur during execution. In particular, users are empowered to defer decisions to runtime

where the system may detect and therefore react against unforeseen situations.

– **Phase of decision making:** The phase of decision making determines the stage in which changes can be identified and managed. Design-time decision making deals with the definition of processes and all possible exception handlers at design-time. Such processes, called *structured processes* [7] or *pre-specified processes* [8], after having been modeled, may be repeatedly instantiated and executed in a predictable and controlled manner. However, in scenarios where changes are unclear or, even, if processes are to a large extend unstructured, runtime decision making is required.

With the below figure in mind and referenced to the flexibility needs, BP design-time variability techniques may be regularly applied to static environments in order to deal with large amount of similar processes (i.e., process families). In contrast, when the dynamism of environments grows, there is a need of supporting adaptation and evolution by pre-specified processes. However, where changes are not anticipated and consequently processes are unclear, processes demand a certain degree of looseness. Late Selection pattern might be valid for dynamic business environments where uncertainty is not an extreme issue and BP design-time variability management becomes a rigid solution (e.g., the upper part of the first and second quadrants of Fig. 1). Specifically, from our viewpoint, runtime variability may take advantage when the following conditions are satisfied: (i) large number of instances are required, (ii) fragment selection depends on context information only available at runtime, (iii) process execution may not be interrupted (i.e., system 24/7 availability is required), and (iv) enactment and historical data become crucial for variant selection. Finally, Late Modeling/Composition are better sustained by unpredictable conditions.

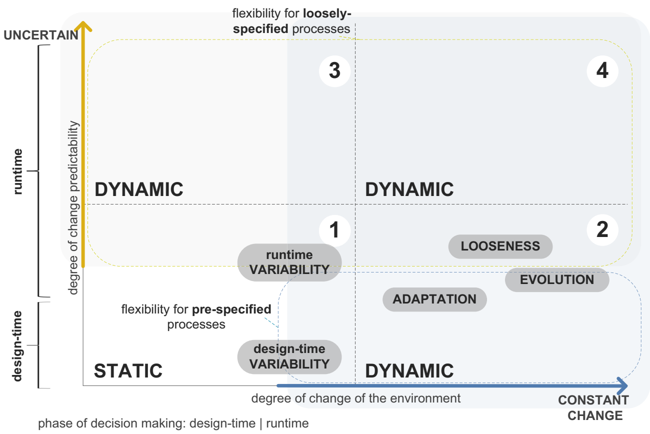


Fig. 1. Dynamic Business Environments Characterization

2.3 Problem Statement

Focusing on Late Selection pattern, it clearly separates modeling (design-time), in which pre-specified parts and placeholder activities are defined, and execution (runtime) in which activity selection is allowed. According to [8], decisions on how to concretize a loosely-specified process model during runtime may be made in different ways: *goal-based* where the selection is based on process goals, *rule-based* where the selection is based on pre-defined rules, *user-based decisions* where the selection is done manually, and *experience-based* where decisions may use previous activities' executions or even past instances' execution information. Similarly, the selection of the placeholder implementation can either be *automated* (i.e., the selection is accomplished automatically), *system-supported* (i.e., the selection can be realized interactively by end-users) or *manual* (i.e., the selection is done by end-users).

Despite the previous research efforts in this field, such as the Worklets approach [4], and static process-based service composition approaches [12,13,14], these proposals present some limitations referred to: (i) managing process fragments separately from the base model (i.e., separation of concerns), (ii) deferring the resolution of variation points by selecting pre-defined alternatives automatically based on constraints and context information during runtime, and (iii) making decisions by relying on past experiences made in similar context.

From our perspective, as placeholder activities are resolved during runtime and knowledge is acquired via experience, it is important to consider experience logs in order to select automatically the most appropriate variant given a particular context. Hence, the resolution of the placeholder activity could be different taking past experiences into consideration. This decision making method leverages static pre-defined rules or goals, bringing Late Selection pattern to its full potential for dynamic business environments with a touch of uncertainty.

3 Case Study: Elevators Maintenance

The elevators maintenance plans (e.g., corrective, preventive, and predictive) ensure adjustments and parts that need to be replaced with the aim of extending the working life of elevators. In this context, the *elevators remote predictive maintenance and monitoring process* plays an important role. This process includes a series of tasks in order to ensure operational safety and, if necessary, detect potential failures that could happen in the near future. In the following points, the predictive maintenance and monitoring process of two types of elevators, `ElA` and `ElB`, are described (see Fig. 2).

Example 1. `ElA` and its modules are constructed by LiftCorp company and installed in a detached house of Spain. `ElA` sends data about its modules status (e.g., door and pulley) to a central maintenance and monitoring system in a predefined time frame (every 15 min). Once the central remote maintenance system receives data, it checks the contract type of the particular client. Depending on such maintenance agreement, the predictive maintenance system starts by

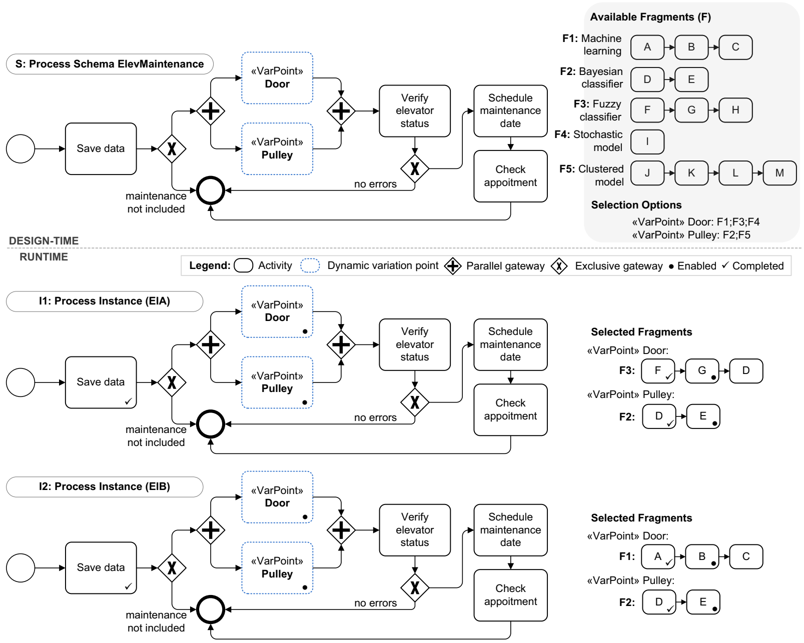


Fig. 2. Illustrative Elevators Maintenance Process

analyzing if each module is working as expected or could be accompanied by any anomaly. In this particular case, the customer has paid for a premium 24h non-stop maintenance, so the support is maximum for elevator's modules, i.e., all maintenance methods for failure prediction are enabled. Each fragment implements a different online failure prediction method with its own particularities (e.g., computation time, level of detail of the analysis, etc.) [15]. After data processing, in this case F3 and F2 are executed, the system detects that the pulley could be broken in a few days, so it calls to the scheduler finding any available maintenance slot. The scheduler finds an adequate maintenance appointment that is sent to the person in charge of elevator's maintenance. Finally, the person in charge validates the proposed maintenance date updating operator's schedule.

Example 2. `ElB` and its pulley are assembled by LiftCorp company but the door is provided by ConveyingCorp company. The elevator is installed in a building of Switzerland. As well as `ElA` elevator, door and pulley status are sent to a central maintenance system (every 20 min). `ElB`'s customer has paid for a basic maintenance, so the spectrum of failure prediction methods for each module is reduced (e.g., F3 is excluded for door predictive maintenance). Once data is processed, in this case F1 and F2 are executed, the system detects that the door needs further maintenance so calls the scheduler to find an appropriate revision date given the available resources from ConveyingCorp company and customer's contract type. After that, the proposed maintenance date is sent to LiftCorp supervisors to verify that really `ElB` requires for an inspection. The supervisors validate the date so the task is assigned to a ConveyingCorp operator.

As extracted from the examples above, the remote predictive maintenance and monitoring system has to consider different types of elevators, modules, maintenance contracts, and operators, therefore variability should be managed. In this particular case, as illustrated in Fig. 2, the selection of a placeholder activity (in boxes with `VarPoint`, colored in blue) should be made just-in-time based on current context information (i.e., the specific data sent by an elevator) for selecting an appropriate online failure prediction method exposed as fragment for each module type. In addition, elevators remote predictive maintenance and monitoring service could provide users with a machine that runs at peak performance 24/7, so it is seemingly inconceivable to interrupt the system. Finally, selection of applicable process fragments for modules' maintenance could make use of historical data in order to select the best variant given the actual context information. Under these conditions, variability should be treated and managed at runtime.

4 The LateVa Approach

LateVa (**Late Va**riability of Process Models through Automated Selection of Fragments) supports the different phases of the BP life-cycle [16], i.e., *modeling* by the *Process Modeler*, *execution* by the *Runtime Engine*, and *evaluation* by the *Process Explorer*. Fig. 3 shows the LateVa framework architecture and it also points out the relationship among the implementation of the different framework components in our prototype platform. The core components of the framework technology are independent providing a decoupled integration solution.

During the modeling phase, the *Base Model*, the *Variation Model* and fragments can be validated against inconsistencies and errors, and simulated to ensure the right execution before deployment to the *Models Repository*. In this case, the *Resolution Model* creation is performed during process execution. The particularities of each component are described in the following subsections.

4.1 Process Modeler

The *Process Modeler* allows for the development of reference process models (i.e., process schemes) with the corresponding variation points and process fragments (i.e., variation point implementations). To that end, we make use of the Base-Variation-Resolution (BVR) modeling approach from the Software Product Line Engineering (SPLE), which clearly states the separation of model commonalities, possible variants, and their configurations in different models [17]. The authors basically defined three different models for variability modeling: a *base model* (representing commonalities), a *variation model* (representing individualities), and a *resolution model* (representing model configurations). Hence, variability is not explicitly applied to individual Domain Specific Languages (DSLs) (e.g., Business Process Modeling Notation (BPMN)), so the base model is not overloaded with variant specific information.

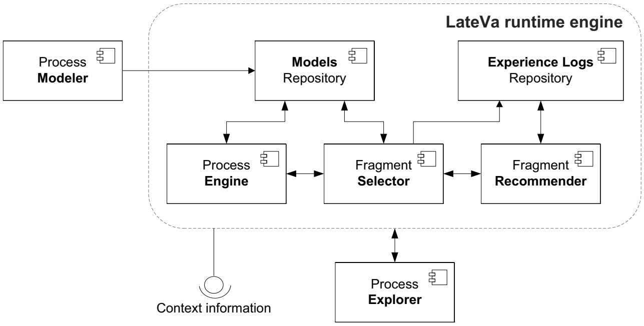


Fig. 3. The Architecture of LateVa

Base Model. The base model describes, using a BP modeling language (e.g., BPMN), the commonalities shared by all model variants and placeholder activities that are subjected to vary. Placeholder activities (i.e., variation points) identify process parts that contain variability and in which variant binding occurs, as illustrated in Fig. 4. In formal terms, a base model is defined as:

Definition 1. *Base Model $\mathcal{BM}$ is defined as a 2-tuple $<\mathcal{C}, \mathcal{VP}>$, being $\mathcal{C}$ the set of commonalities and $\mathcal{VP}$ the set of possible variation points. Every process schema $\mathcal{S}$ derived from $\mathcal{BM}$ consists of a subset of commonalities and a subset of variation points: $\forall\ \mathcal{S} \in \mathcal{BM}$, $\mathcal{S} = (\{c/c \in \mathcal{C}\}, \{vp/vp \in \mathcal{VP}\})$.*

The resolution of variation points (i.e., binding) may be solved either at *design-time* (when the selection of a particular implementation for the placeholder activity depends on initial static preferences) or at *runtime* (when the selection for a particular implementation for the placeholder activity depends on dynamic context information of a process instance). Late Selection pattern is focused on the latter where only parts of the process are pre-specified at build-time, while the implementation for the actual placeholder activity can be concretized during runtime by selecting it from a set of available fragments. Considering context information, LateVa is capable of dynamically binding variations points at runtime, as in Dynamic Software Product Lines (DSPLs) [18,19].

Albeit an unique base model with N variation points and M fragments may serve to support runtime variability of multiple instances, performance overhead exponentially may grow when a certain threshold of variation points (N) or fragments (M) is reached, which clearly impacts scalability. Thereby, we have also considered a partial resolution of the primary base model extracting a pattern with a small number of fragments inspired by [20], i.e., simplifying the spectrum of fragment choices for runtime.

Variation Model. The variation model gathers all the individualities introduced by each process model variant describing individualities and constraints. Individualities are characterized by process fragments which in turn are related

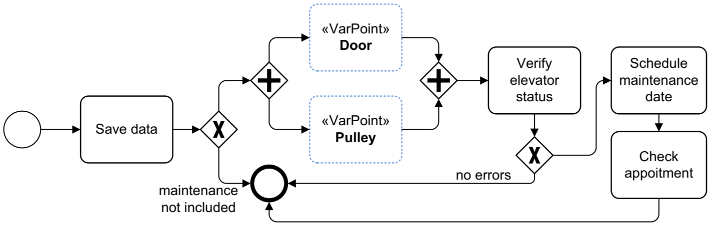


Fig. 4. Base Model

to one or more variation points (see Fig. 5). Thus, each variation point may have a number of possible implementations with nested variation points. These relationships are subjected to changes, since new fragments may be added during process instance execution. A variation model is formalized as follows:

Definition 2. (Variation Model) *Variation Model $\mathcal{VM}$ is defined as a 4-tuple <$\mathcal{P}$, $\mathcal{VP}$, $\mathcal{F}$, $\mathcal{O}$>, being $\mathcal{P}$ the variability sub-model made up of the set of features, $\mathcal{VP}$ the set of variation points of a base model, $\mathcal{F}$ the set of possible applicable fragments, and $\mathcal{O}$ the constraint sub-model made up of features and fragment constraints.*

Variability Sub-Model. The variability sub-model is characterized by two types of variability representations: *feature variability* and *product realization variability.* The former, just like in feature modeling, represents the features present in a particular domain as (mandatory/optional feature, dependencies, XOR/OR, and cardinality). The latter, in turn, defines relationships between variation points described in a base model and possible applicable fragments.

Constraint Sub-Model. As well as in the variability sub-model, constraints may be defined at two levels: *dependencies among features* and *constraints among fragments.* While the former specifies interdependencies of the features of the domain model, the latter represents the dependencies among different placeholder activities implementations (i.e., fragments) of different variation points. Concerning to feature constraints, those may be *hard* (i.e., they are to be necessarily satisfied) or *soft* (i.e., only express a preference of some solution, and so there is not an obligation to fulfill them). Thus, if a feature constraint is hard, the corresponding fragment have to be selected. On the other hand, the selection of one fragment may be directly influenced by another fragment selection. Two types of relations may be considered in this part: *requires* and *excludes.* The requires relationship is a relation defined between two or more fragments where the instantiation of one of them in a specific variation point forces the instantiation of the related fragments in others variation points. In contrast, the excludes relationship is defined as a relation between two or more fragments where the instantiation of one of them in a specific variation point forces the no instantiation of the related fragments in other variation points.

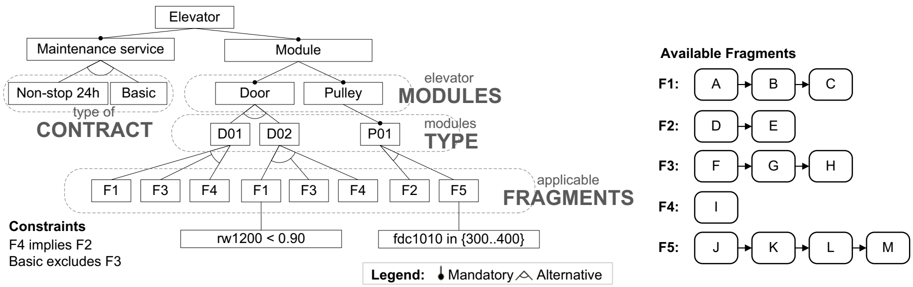


Fig. 5. Variation Model (hard/soft constraints are omitted)

Resolution Model. The resolution model specifies which alternatives described within the variation model are valid for placed placeholder activities in the base model given a particular context (see Fig. 6). A resolution model is formalized as follows:

Definition 3. (Resolution Model) *Resolution Model $\mathcal{RM}$ is defined as a 3-tuple $<\mathcal{CV}, \mathcal{V}, \mathcal{RF}>$, being $\mathcal{CV}$ the set of context variables, $\mathcal{V}$ the set of context values, and $\mathcal{RF}$ the set of relevant fragments.*

This model consists of two main blocks: *context information* representing conditions (CV and V) and *relevant fragments* representing conclusions.

Context Information. Context information details conditions that must be satisfied to select a desired fragment. Conditions are specified by determining which context variables have to be evaluated and their values. Two types of context variables are distinguished: *static* context variables and *dynamic* context variables. The big difference between them is that dynamic variables are only available when the selection of a placeholder activity occurs (i.e., just-in-time), while static variables are known just before the process instantiation.

Relevant Fragments. Fragments are associated with the resolution of variation points depending on the given context information. If a set of context variables are fulfilled then the fragment indicated for the resolution of that variation point will be invoked.

4.2 Runtime Engine

The *Runtime Engine* is the heart of LateVa. It is a process engine that runs process models and invokes the *Fragment Selector* and the *Fragment Recommender* in order to resolve variation points using an automated reasoning of features or even a process mining mechanism depending on the received context data.

Context and Fragment Selection. The *Process Engine* monitors new process instance requests. Based on this data, the resolution of variation points is performed in two stages, as illustrated in Fig. 7.

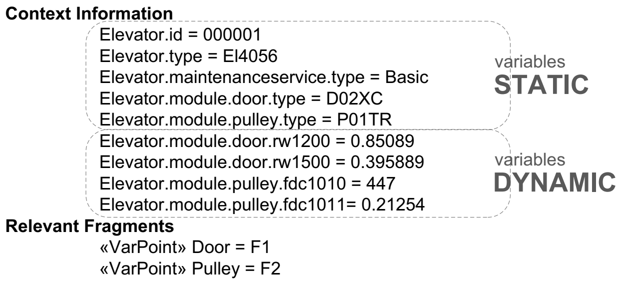


Fig. 6. Resolution Model for ElB

The first stage deals with static preferences to conclude which fragments are still applicable after process instantiation (e.g., F3 is excluded for door due to basic maintenance contract). LateVa uses an automated reasoning mechanism for that purpose. It transforms variation model specification into a Constraint Satisfaction Problem (CSP), inspired by [21], in which different solutions may be found: (i) just one solution with no preferences, (ii) all solutions, (iii) an optimal solution by means of an objective function defined in terms of one or more context variables. As a result, if all variation points are not resolved, the *Fragment Selector* creates a partially resolved resolution model in the *Model Repository* and the *Process Engine* starts with process instance execution.

In the second stage, the *Fragment Selector* also employs a CSP to conclude which fragments are satisfied by dynamic data. Compared to the previous phase, selection starts just before placeholder activity execution. At this precise point in time, a comprehensive analysis of hard/soft constraints and constraints among fragments is accomplished to deduce a convenient fragment.

If a solution is sought (e.g., derived from a set of constraints. A CSP contains a set of constraints (hard and soft) which model the domain of variables and restrictions that have to be fulfilled), the resolution model is completed by setting up a link to a fragment implementation (e.g., F2 is selected for pulley due to *"fdc1010 in 300..400"* constraint). In case of more than one solutions are applicable, the fragment selection is delegated to the *Fragment Recommender* looking at *Experience Logs Repository* for suggestions of all similar conditions.

Selection Aided by a Fragment Recommender System. The *Fragment Recommender System* uses process mining based on historic information in *Experience Logs Repository* to provide operational decision support. Process mining may be used to check conformance, predict the future, and recommend appropriate actions [22]. In this particular case, LateVa focuses on process instances (cases) that have not yet completed and uses process mining to recommend possible applicable fragments on the basis of actual context information.

Fragment logs are stored in the repository for each variation point resolution. Nevertheless, if variation points can be solved by more than one variant (extracted by the *Fragment Selector*), the *Fragment Recommender* makes use of

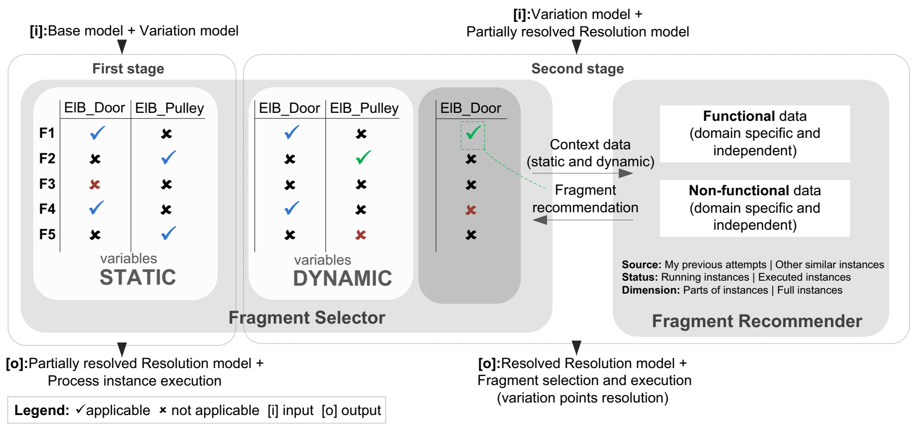


Fig. 7. Runtime Two-staged Fragment Selection for ElB

historic *functional* and *non-functional* data in order to suggest a fragment given a desired behavior (e.g., F1 is recommended for door predictive maintenance based on current context data). Users are empowered to configure the *Fragment Recommender* by adjusting *source*, *status*, and *dimension* properties. The reply from the recommender is forwarded to the *Process Engine* which continues with process instance execution. Thus, LateVa provides a system capable of solving automatically variation points depending on context information.

4.3 Process Explorer

LateVa *Process Explorer* provides access to the *Runtime Engine* for process administrators. It includes three key functionalities:

1. *Variability Management*: This module offers the possibility to manage and change pre-defined variation sub-models. Variability and constraint models are altered in order to customize system behavior by changing constraints and adding new fragments to the variation model. This is one of the main advantages of applying Late Selection pattern, i.e., the system may adapt its functionality by adding/changing fragments.
2. *Instance Inspection*: Basically, process administrators should be able to inspect and filter details of running process instances by accessing this module.
3. *Statistical Historic Data Inspection*: This module allows for managing historic data and customizing recommender properties. It is important to supervise historical data to ensure proper system operation.

5 Related Work

The use of Software Product Lines (SPLs) for BPM has been successfully applied during the last decade [23]. Regarding *process variability*, there exists a number of approaches mainly focused on design-time variability management [24]. VxBPEL [25] enables to capture variation points, variants, and realization relations between variation points within a Business Process Execution Language (BPEL) process extending ActiveBPEL for variability support in service-centric systems. The Provop approach [26] was motivated by the fact that "process variant can be created by adjusting a given process model to a given context". It gathers process context by means of process variables and also manages separately process fragments from the base model.

Similarly, the BVR approach has been transferred to the BPM field for managing variability of process model variants [27], using BPMN. In this particular case, once process variant instances are running, authors employ MoRE-BP engine to adapt and evolve running instances to a new base model definition depending on context data. Baresi et al. [28] present an analogous approach based on CVL and BPEL for managing process reconfiguration at runtime. This approach uses DyBPEL which enables the adaptation of BPEL processes. These proposals has been focused on providing a variability management solution to handle BP variants at design-time, however, we make use of the BVR approach to bind variation points at runtime.

Previous research efforts have applied different engineering techniques to deal with BP *runtime variability though employing Late Selection pattern.* In general, they can be classified into two categories. The first category consist of work that aims only at the BP layer. The work that is closer to our proposal is the one presented in [4]. Here, the authors presents the Worklets approach which enables a dynamic runtime selection of self-contained sub-processes aligned to each activity depending on the context of the particular instance. Hence, when activities become enabled the selection of the appropriate fragment is achieved through the use of Ripple Down Rules (RDR), which include a hierarchically organized selection rules. However, in this work, the selection of sub-processes does not take historical data into account and even the selection is realized interactively by end-users.

Work in the second category focuses on static process-based Web service composition within the service layer [12,13,14]. For instance, the Discorso framework [14] offers a comprehensive service-based solution for specifying and managing flexible and responsive business processes. At runtime, the framework oversees the enactment of the different activities' execution and responds to failures or Quality of Service (QoS) violations enabling the QoS-based service selection. As well as Discorso, Canfora et al. [12] propose a QoS-aware binding approach, based on Genetic Algorithms, which supports runtime rebinding whenever QoS estimations deviate. Similarly, concerning QoS properties, process instance self-management capabilities are introduced by aspect-orientation in [13], using AO4BPEL. In those approaches, decision making method leverages static pre-defined rules or goals, leaving experience-based selection aside.

Automated reasoning of feature models have been proposed by [21], but LateVa employs a CSP for selecting the suitable fragment at runtime given a particular context. *Process mining* based on historic information hidden in logs is used in various cases. Aalst et al. [22] provide the ProM tool for time-based operational support to predict the remaining processing time and recommend activities that minimize flow times. In [29], context-sensitive process recommendations based on the analysis of user behavior, crowd processes, and continuous application of process detection are provided. In this paper, process mining is used to recommend possible applicable fragments.

6 Conclusions and Future Work

Managing process variability could bring benefit to both multiple stakeholders working in dynamic business environments and process designers who are dealing with large collections of process variants. From our analysis of process flexibility, dynamic business environments and industrial case study, we have introduced the foundations of the LateVa approach to provide an end-to-end solution for BP runtime variability management. LateVa improves BP runtime variability management by employing CSP and process mining for selecting automatically relevant fragments from a set of available ones. Moreover, our novel approach has practical implication for BP runtime variability management and development by means of: (i) separation of concerns in BP variability management, (ii) automatically selecting process fragments using CSP and context data, and (iii) making decisions by relying on past experiences made in similar context.

In our future work, we will concentrate on developing a tool to provide an end-to-end solution for BP runtime variability management. Furthermore, we are working on the LateVa modeler by integrating the Common Variability Language (CVL)[1] with Activiti BPMN2 Engine[2], and we are also involved in the runtime engine to support dynamic process variability execution and optimization.

References

1. Weber, B., Sadiq, S., Reichert, M.: Beyond rigidity - dynamic process lifecycle support. Computer Science - Research and Development 23(2), 47–65 (2009)
2. van der Aalst, W.M.P.: Business Process Management: A Comprehensive Survey. ISRN Software Engineering (2013)
3. Lanz, A., Reichert, M., Dadam, P.: Robust and Flexible Error Handling in the AristaFlow BPM Suite. In: Soffer, P., Proper, E. (eds.) CAiSE Forum 2010. LNBIP, vol. 72, pp. 174–189. Springer, Heidelberg (2011)
4. Adams, M., ter Hofstede, A.H.M., Edmond, D., van der Aalst, W.M.P.: Worklets: A service-oriented implementation of dynamic flexibility in workflows. In: Meersman, R., Tari, Z. (eds.) OTM 2006. LNCS, vol. 4275, pp. 291–308. Springer, Heidelberg (2006)
5. Pesic, M., Schonenberg, M.H., Sidorova, N., van der Aalst, W.M.P.: Constraint-based workflow models: change made easy. In: Meersman, R., Tari, Z. (eds.) OTM 2007, Part I. LNCS, vol. 4803, pp. 77–94. Springer, Heidelberg (2007)
6. Weber, B., Pinggera, J., Zugal, S., Wild, W.: Alaska simulator toolset for conducting controlled experiments on process flexibility. Inf. Systems Evolution (2011)
7. Kemsley, S.: The changing nature of work: from structured to unstructured, from controlled to social. In: Rinderle-Ma, S., Toumani, F., Wolf, K. (eds.) BPM 2011. LNCS, vol. 6896, p. 2. Springer, Heidelberg (2011)

[1] http://variabilitymodeling.org

[2] http://activiti.org

8. Reichert, M., Weber, B.: Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies. Springer (2012)
9. Schonenberg, M.H., Mans, R., Russell, N., Mulyar, N., van der Aalst, W.M.P.: Process flexibility: a survey of contemporary approaches. In: Dietz, J.L.G., Albani, A., Barjis, J. (eds.) CIAO! 2008 and EOMAS 2008. LNBIP, vol. 10, pp. 16–30. Springer, Heidelberg (2008)
10. Rosemann, M., Flender, C.: Contextualization of business processes. International Journal of Business Process Integration and Management (2008)
11. Reichert, M.: What BPM Technology Can Do for Healthcare Process Support. In: Peleg, M., Lavrač, N., Combi, C. (eds.) AIME 2011. LNCS, vol. 6747, pp. 2–13. Springer, Heidelberg (2011)
12. Canfora, G., Penta, M.D., Esposito, R., Villani, M.L.: A framework for QoS-aware binding and re-binding of composite web services. JSS 81(10), 1754–1769 (2008)
13. Charfi, A., Dinkelaker, T., Mezini, M.: A plug-in architecture for self-adaptive Web service compositions. In: ICWS (July 2009)
14. Ardagna, D., Baresi, L., Comai, S., Pernici, B., Comuzzi, M.: A service-based framework for flexible business processes. IEEE Software 28(2), 61–67 (2011)
15. Salfner, F., Lenk, M., Malek, M.: A Survey of Online Failure Prediction Methods. CSUR 42(3) (2010)
16. Weske, M.: Business process management: concepts, languages, architectures. Springer (2007)
17. Bayer, J., Gérard, S., Haugen, O., Mansell, J., Moller-Pedersen, B., Oldevik, J., Tessier, P., Thibault, J., Widen, T.: Consolidated product line variability modeling. In: SPLC (2006)
18. Hallsteinsen, S., Hinchey, M., Park, S., Schmid, K.: Dynamic software product lines. IEEE Computer 41(4), 93–95 (2008)
19. Bencomo, N., Hallsteinsen, S., Santana de Almeida, E.: A View of the Dynamic Software Product Line Landscape. IEEE Computer 45(10), 36–41 (2012)
20. Czarnecki, K., Helsen, S.: Staged Configuration Using Feature Models. In: Nord, R.L. (ed.) SPLC 2004. LNCS, vol. 3154, pp. 266–283. Springer, Heidelberg (2004)
21. Benavides, D., Trinidad, P., Ruiz-Cortés, A.: Automated Reasoning on Feature Models. In: Pastor, Ó., Falcão e Cunha, J. (eds.) CAiSE 2005. LNCS, vol. 3520, pp. 491–503. Springer, Heidelberg (2005)
22. van der Aalst, W.M.P., Pesic, M., Song, M.: Beyond process mining: From the past to present and future. In: Pernici, B. (ed.) CAiSE 2010. LNCS, vol. 6051, pp. 38–52. Springer, Heidelberg (2010)
23. Santos Rocha, R.D., Fantinato, M.: The use of software product lines for business process management: A systematic literature review. Information and Software Technology (2013)
24. Valença, G., Alves, C., Alves, V., Niu, N.: A Systematic Mapping Study On Business Process Variability. IJCSIT 5(1) (2013)
25. Koning, M., Sun, C., Sinnema, M., Avgeriou, P.: VxBPEL: supporting variability for Web services in BPEL. Information and Software Technology 51(2) (2009)
26. Hallerbach, A., Bauer, T., Reichert, M.: Capturing Variability in Business Process Models: The Provop Approach. Software Maintenance and Evolution 22(6-7), 519–546 (2010)
27. Ayora, C., Torres, V., Pelechano, V., Alférez, G.H.: Applying CVL to Business Process Variability Management. In: VARY (2012)
28. Baresi, L., Guinea, S., Pasquale, L.: Service-Oriented Dynamic Software Product Lines. IEEE Computer 45(10), 42–48 (2012)
29. Burkhart, T., Werth, D., Loos, P., Dorn, C.: A Flexible Approach Towards Self Adapting Process Recommendations. Computing and Informatics 30(1) (2011)

Towards Highly Adaptive Data-Intensive Systems: A Research Agenda

Marco Mori* and Anthony Cleve

PReCISE Research Center, University of Namur
Rue Grandgagnage 21, 5000 Namur, Belgium
{marco.mori,anthony.cleve}@unamur.be

Abstract. Data-intensive software systems work in different contexts for different users with the aim of supporting heterogeneous tasks in heterogeneous environments. Most of the operations carried out by data-intensive systems are interactions with data. Managing these complex systems means focusing the attention to the huge amount of data that have to be managed despite limited capacity devices where data are accessed. This rises the need of introducing adaptivity in accessing data as the key element for data-intensive systems to become reality. Currently, these systems are not supported during their lifecycle by a complete process starting from design to implementation and execution while taking into account the variability of accessing data. In this paper, we introduce the notion of data-intensive self-adaptive (DISA) systems as data-intensive systems able to perform context-dependent data accesses. We define a classification framework for adaptation and we identify the key challenges for managing the complete lifecycle of DISA systems. For each problem we envisage a possible solution and we present the technological support for an integrated implementation.

Keywords: data-intensive systems lifecycle, context-aware database, self-adaptive systems.

1 Introduction

Data-intensive systems manage complex and huge amount of data that are suited for different types of users each performing tasks of different nature and possibly in different contexts. Most of the effort for designing, maintaining and evolving these systems depends on their complex interactions with big data sources. Taking this perspective, a relevant problem that need to be tackled in a systematic manner is how to ease the management of their variability in accessing data. This problem has started to be tacked in the literature of *context-aware databases* [1] by means of methodologies, techniques and tools for creating sub-portions of a global database based on different factors, i.e., current context, user tasks and user preferences [22,7]. These techniques support variations of data that are not

* Beneficiary of an FSR Incoming Post-doctoral Fellowship of the *Académie universitaire 'Louvain'*, co-funded by the Marie Curie Actions of the European Commission.

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 386–401, 2013.

performed in a systematic process where context-dependent variations of application behavior are propagated to data thus making it difficult to consistently change data.

In order to generate the data of interest at a certain context we should reason on the requirements that should be achieved in that specific context [38]. Through a systematic approach it should be possible to create context-dependent versions of data-related artifacts that will remain un-changed for the whole system lifetime. This first level of data-adaptivity (design-time), represents a possible solution which is suited for all the situations where there is no need of reconfiguring data and programs at run-time. Nevertheless, this solution becomes un-appropriate in ubiquitous environments that are characterized by every-changing contexts leading to a two-fold problem. First, the set of contexts can be too large in terms of tasks that will be completed, different types of users accessing the same system and different situations in which the system will have to operate. Second, limited capacity devices may not be able to manage the needed big data source. Thus a second level of adaptivity (run-time) is needed for achieving reconfigurations of data-related artifacts in a systematic and continuous manner. In the literature of *self-adaptive systems* [34,5,17] different techniques support context-dependent behavioral adaptations both at design time and at run-time while not much attention has been devoted to data-manipulation adaptations. Data-intensive systems can benefit from these techniques to achieve design-time and run-time adaptations of data-manipulation programs while they can benefit from *context-aware database* approaches for propagating these variations to data-related artifacts.

Variability of accessing data poses the interest towards a new class of *data-intensive self-adaptive (DISA) systems* as systems able to ease the complexity of accessing data by means of creating context-dependent data-related artifacts at design time and by means of enabling their run-time reconfigurations. In this paper, we envisage a lifecycle process for DISA systems and we propose a research agenda organized according to the three main challenges that need to be addressed in order for these systems to become a reality; we consider a *design* process for DISA systems, *migration* of existing systems towards DISA systems and *monitoring/optimizing* run-time reconfigurations of DISA systems. In [24] we have defined a theoretical framework for supporting the design and configuration of new DISA systems. In this paper, we extend our previous results in a wider scope and we analyze the key challenges for DISA systems along with our methodological solutions for each of those. Beyond the creation of new systems, it is also important to consider legacy data-intensive systems. These systems need to be analyzed in order to evaluate their variability in accessing data. To understand this variability, it is necessary to extract their data-manipulation behavior by analyzing the system execution. Thus we consider the problem of process understanding in order to evaluate the convenience of migrating towards DISA systems. Finally, a migrated or a new DISA system have to be provided with a decision-making process to support its optimized run-time reconfiguration of programs and data-related artifacts.

We consider an e-health scenario where physicians, i.e, doctors, secretaries, nurses, radiologists and patients are involved in a set of care processes. Each physician is interested in a different excerpt of data. Secretaries are interested in administrative data, doctors are interested in case histories (with medical images) of patients, diagnosis and therapies, nurses and patients are interested in the application of the therapy while radiologists are strictly interested in case histories and basic patient information with the aim of capturing medical images. Further, there exist other factors that affect the portion of data of interest, namely the task the physician is performing, the device, the location or room where the system will run and the department to which the patient belongs. For instance, doctors access to high quality medical images only through desktop devices. They access to different set of information based on the activity they perform, i.e., check-up, visits, surgery operations and department administrator. Finally, if the doctor performs a visit from outside the hospital or in case of emergency, he should only visualize a textual representation of the case history. This scenario shows the context-dependent interest of users towards heterogeneous data. The application can be provided with the required data once for all the system lifetime or it may be necessary to reconfigure at run-time the data due to context variations.

In the remainder of this paper, Section 2 gives a detailed descriptions of DISA artifacts and a possible classification of data adaptations. Section 3 analyzes the three main challenges for supporting the lifecycle of DISA systems along with an integrated set of techniques to be adopted for implementing the process. For each problem we present contributions in the literature and our methodological approach. Finally, Section 4 discusses related work before conclusions and future directions are given in Section 5.

2 Framework Basics

In this paper we adopt a feature engineering perspective [19,37,11] in order to represent the basic unit of behavior of DISA systems as *features*. In these systems most of the functionalities operates on data, thus we consider their corresponding features for our analysis. Section 2.1 discusses artifacts of a DISA system while Section 2.2 classifies adaptations to data-related artifacts.

2.1 DISA Artifacts

Fig. 1 envisages the relationships between the artifacts of DISA systems, namely *context*, *features* and *data*. *Context* is characterized by means of a set of dimensions determining the current user situations, namely, user role, user task, device characteristics, location etc... Context states determines the set of features, i.e., *configurations*, that have to included into the application. Configurations defined according to a *feature model* [37] require a subset of data belonging to a big data source suited for all possible contexts. We consider different levels of abstraction of data: the *conceptual schema* is a Platform Independent Model (PIM) typically represented through an ER diagram containing entity types and relationships among entity types; the *logical schema* is a Platform Specific Model (PSM) containing tables and foreign keys which are the basis for defining database queries; finally *database instances* are data to be loaded into the device.

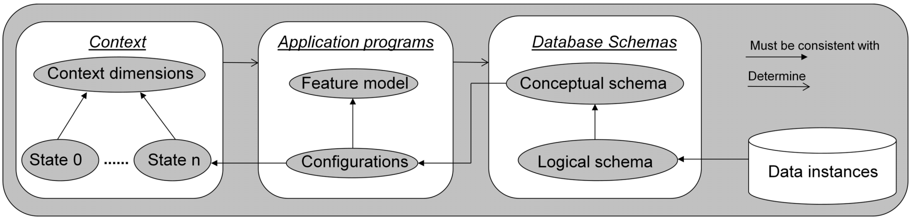


Fig. 1. DISA artifacts

In our perspective a *feature* links together variability of the application (in terms of its requirements) to the variability of data (in terms of data excerpts) as inspired by approaches where variability of requirements is linked to variability of a formal specification [8] or to the variability of source code [16]. Following the taxonomy proposed in [14] we distinguish among functional, performance and specific quality requirements which pertain to functional, performance and specific quality concerns of the application. Orthogonally to these requirements, constraints limit the solution space of functional, performance and specific quality requirements. Based on this taxomony we define a feature as triple $f = (R, P, V)$ where R is a functional, non-functional or a specific quality requirement (context independent), P is the presence condition, i.e. a contextual constraint requirement which expresses the applicability of the feature; V is the excerpt of data of interest for the feature defined in terms of entity types of the conceptual schema.

2.2 Data Adaptations Dimensions

In this section we present a classification of data adaptations based on which we characterize the adaptations we enable in our framework (Section 3.1). We define database adaptations according to the three following dimensions:

- *Structural dimension*, concerning modifications to database structures and data instances;
- *Semantic dimension*, i.e., evolution of the semantic of data;
- *Consistency dimension*, addressing the impact of adaptation to the correctness of data-related artifacts.

Structural dimension is concerned with the adaptation of the database structures. This regroups modifications applied to the conceptual, logical, physical schemas and data instances. we can classify database evolutions as:

– *Conceptual modifications* typically translate changes in the functional requirements of the information system into conceptual schema changes.
– *Logical modifications* do not modify the requirements but adapt their platform-dependent implementation in the logical schema.
– *Physical modifications* aim at adapting the physical schema to new or evolving technical requirements, like data access performance.
– *Data modifications* aim at translating variations of the schema to the database instance.

Semantic dimension captures the impact of a given database adaptation scenario on the informational content of the target database. In other words, it aims at indicating whether the adaptation involves: *Semantics-augmenting*, *Semantics-decreasing* or *Semantics-preserving* schema modifications.

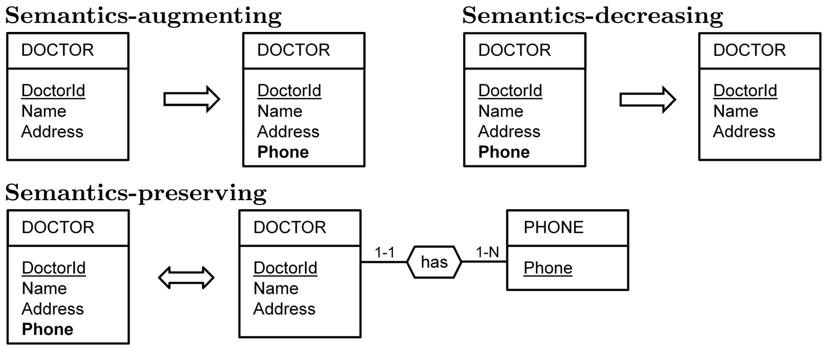


Fig. 2. Categories of schema modifications

Consistency Dimension. Whenever a variation to a data-related artifacts is performed we have to check the consistency of this variation considering two different problems:

– *intra-artifact* consistency: an adaptation may cause an existing consistency constraint to be violated within an artifact.
– *inter-artifact* consistency: a broken consistency link must then be reestablished by means of a *change propagation* adaptation. For instance, schema modifications at a given level of abstraction necessitate the adaptation of the schemas belonging to the other abstraction levels, and the data instances.

3 Variability for DISA Systems

DISA systems have to be created either from zero or from legacy systems taking into account their variability in accessing data. In case of a new system it is necessary to consider which are the factors that tune the interest of the users towards different portions of data and how to propagate variations of this factors to variations of data. DISA systems should support design-time and run-time variability of data. The space of variability is determined at design time while during execution a certain configuration of data is created based on the current context. Only if context changes during the system lifetime we have to provide run-time variability in order to align data variations to context variations. These variations of data that are foreseen at design time are not enough if the system works in an un-predictable environment. Indeed there exist other variations to data that cannot be foreseen before they are needed. Indeed, in ubiquitous environments due to unforeseen situations it is not always possible to exactly determine the space of variability of a system before its execution. To this end, we envisage the adoption of a *design process phase* which is able to support *design-time variability*, *run-time variability* and *unforeseen run-time variability* of accessing data.

In case of the migration of a legacy system to a DISA system, we have to extract its data-manipulation behavior with the aim of understanding its variability in accessing data. This consists of a *process understanding phase* with the aim of evaluating if it is convenient or not to carry out the migration and determining the features of the system as the basis for its variability.

Finally, a DISA system has to perform run-time reconfigurations which are subject to performance degradations and scalability problems that need to be accurately tackled in order for a DISA system to be usable. In the remainder of this section we explain each key problem for a DISA system along with our envisaged solution: (i) *data-variability aware design process*, (ii) *data-variability aware process understanding*, (iii) *data-variability aware performance optimization*.

3.1 Data-Variability Aware Design Process

Context-dependent data accesses should be supported in a systematic manner during the design phase of a DISA system. To this end, it is necessary to align the variability of requirements to the variability of databases by defining traceability links between software functionalities and databases excerpts. Variability of data has been considered in the literature of context-aware databases where many approaches identify the portions of data of interest based on a certain context model describing the current situation. Methodologies, techniques and tools have been defined following either a pruning [42,43,38,36,11] or a merging [2,3,22,7,35,29] technique for creating a subset of a database. Altought it has been argued that it is important to link the variability of the application to variability of data [38], there is no approach that provides the data of interest based on changing application requirements. The literature of self-adaptive systems [34,5,17] provides the theoretical and methodological support for managing the variability of system requirements as a consequence of context variations. On the one hand, the literature of self-adaptive systems provides no support for propagating variations of requirements to data, while on the other hand variability of application requirements has not been considered in the literature of context-aware databases. To this end, taking inspiration by processes provided for self-adaptive systems [4,17], we advice the adoption of a process for designing a DISA system, having as objective the variability of accessing data. We envisage the adoption of a framework that supports feature-based data tailoring by means of a *filtering design process* and a *run-time filtering process*.

The *design process* (Fig. 3) supports the variability of accessing data by establishing the applicability for all the possible feature configurations. It starts by defining the whole set of requirements along with the corresponding global database for all possible contexts. Consequently designers organize the elicited requirements following a feature engineering perspective, through *features* and a *feature model* which entails the admissible configurations of features. At this point designers define a mapping between identified features and portions of the global schema, and they identify the contextual dimensions that affect the interest towards different portion of data. Finally designers define a presence condition for each feature to evaluate if data required by a feature should be included or not in the subset of the database. Finally, the *decision-making support* phase defines the applicability of each admissible configurations at each possible context state based on the presence condition of its entailed features.

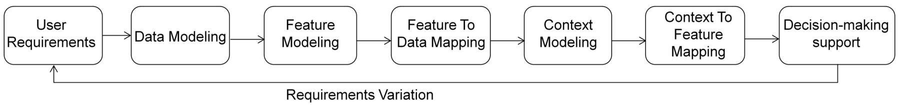
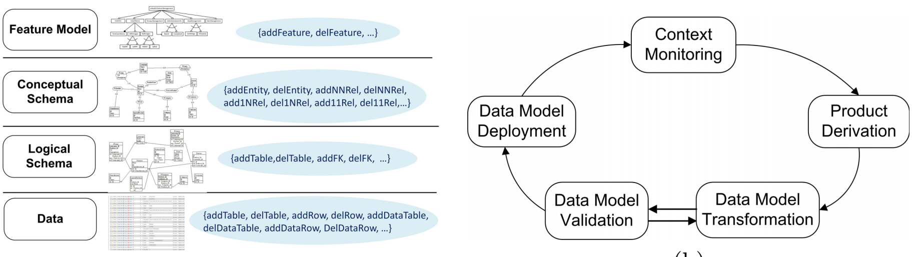


Fig. 3. Feature-based filtering design process

The *run-time filtering process* (Fig. 4(b)) provides the right data according to the features that have to be provided at the current context. Upon context variation an automatic derivation phase retrieves the most *suitable* (see Section 3.3) set of features to apply. According to these feature the process determines the set of entity types of the conceptual schema that should be included in the target view. The *data model validation* phase consists in modifying the conceptual schema in order to make it consistent with the large schema. Once this view has been created the *data model deployment* phase determines the corresponding logical schema and data instances. In Fig. 4(a) we have emphasized the stack for the reconfiguration of data. The target configuration of features is the input for determining the subset of the conceptual schema, which in turn is the input for determining the subset of the logical schema, which in turn determines the actual data. API's provided at each level support the propagation of variation from high-level conceptual schema till to data instances.

Designers should create DISA applications along with their context-dependent variability (Fig. 3) before putting the system in execution. At run-time based on the current context and user role, an automatic procedure should provide the subset of data of interest to the application. If context and user role remain the same for all the system lifetime we have only *design-time variability* of accessing data. On the contrary, if either user role or context change during the system lifetime, we need *run-time variability* of accessing data. In this case an automatic



(a) (b)

Fig. 4. Feature-based (a) reconfiguration stack and (b) run-time filtering process

procedure should enable the continuous run-time variations of data according to continuous context and user role variations (Fig. 4(b)).

This clearly distinction from design-time and run-time activities does not hold in ubiquitous environments which are mainly characterized by variations of requirements due to un-predictable contexts. In this case, the variability space determined at design time may have to be re-computed at run-time taking into account the variations to the requirement set to satisfy. The process we have envisaged in [24] is only able to support variations that are foreseen at design time. To this end, we enhance the process by enabling the *unforeseen run-time variability* of accessing data by means of re-iterating the steps of the process at run-time in order to satisfy a new set of requirements (Fig. 3). We consider only addition of requirements while we do not take into account deletion since the first poses the more difficult problems.

Feature Evolution Scenarios. New requirements encapsulated into new features should be included into the data-intensive systems at run-time. We classify two co-evolution scenarios representing the addition of a new feature:

– Co-evolution between *requirements* and *context*: requirements of the application have to be provided based on the current context. On the one hand, whenever a new requirement has to be implemented into the system it may be necessary to modify the context. For instance, if the hospital buys a new machine for the radiologist department it may be necessary to add a new requirement to acquire a new kind of medical images. This implies the addition of a new applicability condition over the context to define when this requirement should be provided. Thus the context model has to be augmented with the new room where this new kind of images will be collected. On the other hand, whenever the dimensions of context are augmented, it may be necessary to modify the set of requirements. For instance if the set of user roles is augmented with the psychologist (context variation), we may have to consider a new requirement to support his activity.
– Co-evolution between *requirements* and *data*: requirements may require data in order to be fulfilled, thus they may co-evolve each other. Following the same example above, the new requirement to add for collecting new medical

images requires new data portions where the images will be recorded. Thus data schemas and data instances have to be modified in order to include this new type of information. On the contrary, whenever data are evolved (e.g., augmented), new requirements may be required for accessing these data. Let us suppose to add to data the portion corresponding to the long hospital treatments. As a consequence we will have to add a new requirement to allow visualization of this new kind of treatments through the graphical unit interface (GUI).

Data Adaptations in Our Framework. In our framework we envisage the adoption of *conceptual*, *logical* and *data instances* modifications that are *semantics-decreasing* and *semantics-increasing*. In particular, in case of design-time variability we envisage the application of *semantics-decreasing* adaptations in order to produce a subset of information from the global source schema that are suited for a certain context. In case of run-time variability we envisage the adoption of *semantics-preserving* and *semantics-increasing* adaptations since switching from a context to another it is necessary to discard some information that are not anymore required while it is necessary to add other information. Both for design-time and run-time variability we envisage the adoption of *conceptual* modifications in order to align the platform-independent schema to the current set of requirements that have to be provided in a certain context. Consequently, we consider *logical* modifications in order to align the conceptual modification to the platform-dependent schema and finally *data instances* modifications for configuring the required database instance.

Data-related artifacts provided for a certain context are subsets of wider artifacts which include data necessary for all possible contexts. Whenever a variation to a data-related artifact is performed we have to check the correctness of this variation considering two different problems. First, we have to check if such variation is consistent with the rules defined for the definition of that specific model (*intra-artifact consistency*). Second, we have to check if the variation to the data-related artifact is consistent with its global source model (*inter-artifact consistency*). We achieve the first by performing variations that are correct by construction while we achieve the second by applying an adjustments phase to changed models [11]. Once a consistent model is obtained we apply bi-directional transformations [12,39] in order to propagate variations from user-centric data models (i.e., the conceptual schema) till to data instances.

3.2 Data-Variability Aware Process Understanding

Extracting the data-manipulation behavior of the application is a complex task which supports designers in understanding a running data-intensive system. These systems carry out frequent interactions with a data source with the aim of fulfilling their requirements. Capturing these interactions is a promising approach for understanding the application behavior and for supporting three different types of activities: *discovering* the application behavior, *checking* the compliance of the application behavior w.r.t. contracts models, *enhancing* the application behavior. The *discovering* phase is required for legacy data-intensive systems for which documentation of processes is not available. Let us consider the e-health application implemented and working in hospital without documentation; for these application it is interesting to produce the behavioral models that are daily performed by physicians in order to document their actual activities. *Checking* the compliance of the application behavior is useful for both legacy and non-legacy data-intensive systems; let us consider an hospital manager which desires to monitor some parameters of the activities performed by third-party companies working at the hospital; managers agree on a written process and they would monitor that this process is correctly performed in reality. *Enhancing* the application behavior is another important activity for legacy and non-legacy data-intensive systems that allows the variation of the process by adding new instances of it, e.g., in a hospital the manager may want to add a new instance for the patient registration process where user is asked off-site to give a judgment of his experience at the hospital.

The activities we have described are at the core of process mining techniques whose aim is to discover, to check and to enhance real processes starting from event logs of information systems [41,40]. Input events logs entail different instances of the same process as they can be recorded during system execution based on different types of information. Many approach presented in the literature [20,31,21,33,26] show the usefulness of process mining techniques for discovering, checking and enhancing processes in real working environments. The approach presented in [26] applies a process mining technique to logs of web service interactions, while approaches presented in [20,31,21] show the usefulness of process mining techniques in healthcare scenarios. Finally, the approach in [33] shows how to adopt a process mining technique to extract processes of wafer scanners to support testing held by a manufacturer. None of these approaches is designed for data-intensive systems. In order to adopt process mining techniques to data-intensive systems we have to determine higher-level data-manipulation functions starting from the interactions of the systems with its data sources. Approaches presented in the literature like [9,10] support the extraction of semantic information starting from sequences of queries and relationships among those. Semantic information collected by following these approaches should be the basis for determining the events of a data-intensive application in terms of high-level data-manipulation functions. Upon the identification of these data-oriented events, we envisage the adoption of classical process mining techniques in order to produce data-oriented processes of the system.

Process mining for DISA systems can support the analysis of processes related to a single user or to a single group of users or it can support the analysis of multi-user of multi-group of users performing the same process. For example in the e-health scenario we may want to analysis the process of check-up visits for all the doctors and check the variability of accessing data for such a process. In addition we may want to analyze a process containing all the activities in which a single doctor is involved, i.e., check-up visits, surgery operations, department administrator activities, etc...

3.3 Data-Variability Aware Performance Optimization

DISA applications need to be reconfigured at run-time as a consequence of context-variations. Reconfigurations of data involve a set of conflicting requirements that should be carefully taken into account by the framework module implementing the variation of data. Among these requirements we have *stability* of data [18,27], i.e., a measure defined as the ratio between variations of data (output) and variation of context (input), *user benefit*, i.e., a metric expressing the satisfaction of the user, and *reconfiguration cost* which is a metric defined over the operations that have to be completed for reconfiguring data. Let us consider the e-heath case study where the doctor changes his task from check-up activity to an emergency activity. This requires a reconfiguration of the database supporting the doctor mobile application in the emergency activity with a restricted set of data. This reconfiguration should be performed by considering the requirements above and in particular given more weight to the reconfiguration cost requirement since in an emergency situation it is better to have a low reconfiguration cost (e.g., very quickly) and low user benefit (e.g., limited patient information) instead of having high reconfiguration cost (e.g., wait too long) and high user benefit (e.g., rich set of patient information).

In Software Engineering conflicting criteria are combined together with different weights in a unique utility function that is optimized [44,32,28]. Approaches presented in the literature of self-adaptive systems show that these optimization approaches can benefit from predictive models of context. Indeed, looking at future context variations affecting the reconfiguration choices, it is possible to achieve better performance of the reconfiguration process. As presented in [25] the authors exploit a probabilistic user preference model for achieving reconfiguration of self-adaptive systems with better performance while in [6] the authors propose an approach for achieving better reconfiguration performance based on a predictive model concerning the availability of contextual resources. Following the idea of these approaches, our aim is to optimize the performance of the reconfiguration process for DISA systems by adopting a predictive process model containing information about current and future data accesses. As shown in Section 3.2, it is possible to extract processes of data accesses starting from historical information as the basis for getting a predictive model. Based on this model, we envisage the adoption of a multi-objective optimization technique with the aim of promoting better performance for the reconfiguration of data. This technique should support the decision-making process by enabling the evaluation of the most *suitable* configuration of data that should be adopted for *design-time* and *run-time adaptations*.

Let us consider the e-health case study where the doctor is changing his activity from visits management to department administrator and let us suppose that he remains department administrator for a short period of time before coming back to his visits. As soon as he becomes administrator, his application has to be reconfigured in order to include the data required for performing the new activity. Nevertheless it is not convenient to discard all the data required for performing the visits since the doctor will soon return to visit patients. To this end, as much as possible data regarding visits should be maintained into the device taking into account the future context variations.

3.4 Techniques for Implementation

The lifecycle process we envisage for DISA systems can benefit from current practice technologies available in the literature. We present features by defining their requirements (e.g. as Linear Time Temporal Logic expressions), context requirements as predicates and data excerpts as sets of entity types of the conceptual schema. We formalize a *SAT problem* (e.g., JaCoP tool[1]) for evaluating the context states in which each configuration of features is admissible according to its context predicate. Since more than one configuration may be admissible at a certain context state, we envisage the adoption of *multi-objective optimization techniques* for selecting the best possible configuration of data. Once such a configuration is identified, we envisage the application of a *filtering technique* for creating the subset of the global conceptual schema suited for the best configuration at the current context [23]. We consider *schema transformation techniques* for making the subset of the conceptual schema consistent with the global schema. Then, through *bi-directional techniques* we propagate conceptual modifications to the logical schema and finally to data instances. We plan to model data-related artifacts with DB-MAIN tool[2] while we plan to adopt MySQL[3] DBMS for data instances. As far as the migration problem is concerned, we plan to adopt *query parsers* (e.g., *JSqlParser*[4]) for extracting semantic information of data-related events to be applied to a *Formal Concept Analysis technique* [13], (e.g., *colibri-java*[5]) with the aim of clustering queries which implement the same high-level data-manipulation function [15]. Once high-level data accesses have been identified, we plan to adopt the de-facto standard for process mining, i.e., *ProM* tool[6] to extract data-oriented processes of legacy data-intensive systems.

4 Related Work

To the best of our knowledge, a lifecycle process that supports creation, migration and optimization of DISA systems has not been yet proposed in the literature. This process should support design-time and run-time adaptivity of accessing data. Most of approaches presented in the literature consider only specific problems within the process and they provide only design-time variability; they follow either a pruning (top-down) or a merging (bottom-up) perspective

[1] http://jacop.osolpro.com
[2] http://www.db-main.be
[3] http://www.mysql.com
[4] http://jsqlparser.sourceforge.net
[5] http://code.google.com/p/colibri-java
[6] http://www.promtools.org

for creating the data of interest (subset) from a global data model. In [2], the authors propose a merging approach for tailoring the logical schema to the current context instance which is modeled separately from the schema. In [7], the authors work on a logical schema for producing the excerpt of data that fits the current context-dependent user preferences. In [42], the authors propose a filtering approach for creating a consistent excerpt of the conceptual schema starting from a required subset of it. In [38], the authors propose a feature-oriented approach for tailoring the data of interest from a conceptual schema. They model the variability of accessing data in terms of features but they do not link this variability to context variations. In [3], the authors present a design technique for creating very small databases from a big data source by considering conceptual and logical schema. Context variations are not taken into account, thus their approach does not support context-dependent run-time variations of data. As far as run-time variability is concerned, the approach presented in [30] shows how to achieve un-predictable variations of the context and how to propagate this variations to the relational database.

Altought adaptivity of data can be achieved by considering artifacts at different abstraction levels, it is not still clear which is the process to follow in order to create, to migrate and to optimize DISA systems that support design-time and run-time adaptivity of accessing data. A link between application variability and data variability has not been yet implemented in the literature making it difficult to propagate variations of context to variations of required data. Most of approaches support design-time variations of data while there is almost no support for foreseen and unforeseen run-time variations; the latter are becoming more and more important given that it is not always possible to provide the complete space of reconfiguration choices at design time.

5 Conclusions

We discussed the critical problems of data-intensive systems and the need of introducing adaptivity to ease the management of big amount of data for which a context-dependent approach makes sense, i.e., different users accessing the same system, heterogeneous environments where the application runs and heterogeneous processes to be completed over data. We presented a unique lifecycle process for DISA systems and we showed how to solve the three critical problems for the management of data variability. We proposed a methodological solution and a possible integrated implementation which exploits techniques presented in the literature. As for future work we will implement our integrated solution for the lifecycle of DISA systems and we will experiment it at a large scale with a real e-health system, e.g., OSCAR database[7] which contains a huge amount of data of interest for differents stakeholders performing heterogeneous care processes in different contexts.

[7] http://www.new.oscarmanual.org

References

1. Bolchini, C., Curino, C., Orsi, G., Quintarelli, E., Rossato, R., Schreiber, F.A., Tanca, L.: And what can context do for data? ACM 52(11), 136–140 (2009)
2. Bolchini, C., Quintarelli, E., Tanca, L.: Carve: Context-aware automatic view definition over relational databases. IS 38(1), 45–67 (2012)
3. Bolchini, C., Schreiber, F.A., Tanca, L.: A methodology for a very small data base design. Inf. Syst. 32(1), 61–82 (2007)
4. Brun, Y., et al.: Engineering self-adaptive systems through feedback loops. In: Cheng, B.H.C., de Lemos, R., Giese, H., Inverardi, P., Magee, J. (eds.) Self-Adaptive Systems. LNCS, vol. 5525, pp. 48–70. Springer, Heidelberg (2009)
5. Federrath, H. (ed.): Designing Privacy Enhancing Technologies. LNCS, vol. 2009. Springer, Heidelberg (2001)
6. Cheng, S.-W., Poladian, V., Garlan, D., Schmerl, B.R.: Improving architecture-based self-adaptation through resource prediction. In: SEFSAS, pp. 71–88 (2009)
7. Ciaccia, P., Torlone, R.: Modeling the propagation of user preferences. In: Jeusfeld, M., Delcambre, L., Ling, T.-W. (eds.) ER 2011. LNCS, vol. 6998, pp. 304–317. Springer, Heidelberg (2011)
8. Classen, A., Heymans, P., Schobbens, P.-Y.: What's in a *feature*: A requirements engineering perspective. In: Fiadeiro, J.L., Inverardi, P. (eds.) FASE 2008. LNCS, vol. 4961, pp. 16–30. Springer, Heidelberg (2008)
9. Cleve, A., Meurisse, J.-R., Hainaut, J.-L.: Database semantics recovery through analysis of dynamic sql statements. J. Data Semantics 15, 130–157 (2011)
10. Cleve, A., Noughi, N., Hainaut, J.-L.: Dynamic program analysis for database reverse engineering. In: Lämmel, R., Saraiva, J., Visser, J. (eds.) GTTSE 2011. LNCS, vol. 7680, pp. 297–321. Springer, Heidelberg (2013)
11. Czarnecki, K., Antkiewicz, M.: Mapping features to models: A template approach based on superimposed variants. In: Glück, R., Lowry, M. (eds.) GPCE 2005. LNCS, vol. 3676, pp. 422–437. Springer, Heidelberg (2005)
12. Czarnecki, K., Foster, J.N., Hu, Z., Lämmel, R., Schürr, A., Terwilliger, J.F.: Bidirectional transformations: A cross-discipline perspective. In: Paige, R.F. (ed.) ICMT 2009. LNCS, vol. 5563, pp. 260–283. Springer, Heidelberg (2009)
13. Ganter, B., Wille, R., Wille, R.: Formal concept analysis. Springer, Berlin (1999)
14. Glinz, M.: On non-functional requirements. In: RE, pp. 21–26 (2007)
15. Grosso, C.D., Penta, M.D., de Guzmán, I.G.R.: An approach for mining services in database oriented applications. In: CSMR, pp. 287–296 (2007)
16. Inverardi, P., Mori, M.: Model checking requirements at run-time in adaptive systems. In: ASAS 2011, pp. 5–9 (2011)
17. Inverardi, P., Mori, M.: A software lifecycle process to support consistent evolutions. In: de Lemos, R., Giese, H., Müller, H.A., Shaw, M. (eds.) Self-Adaptive Systems. LNCS, vol. 7475, pp. 239–264. Springer, Heidelberg (2013)
18. Karsai, G., Ledeczi, A., Sztipanovits, J., Peceli, G., Simon, G., Kovacshazy, T.: An approach to self-adaptive software based on supervisory control. In: Laddaga, R., Shrobe, H.E., Robertson, P. (eds.) IWSAS 2001. LNCS, vol. 2614, pp. 24–38. Springer, Heidelberg (2003)
19. Keck, D.O., Kühn, P.J.: The feature and service interaction problem in telecommunications systems. a survey. IEEE TSE 24(10), 779–796 (1998)

20. Mans, R.S., van der Aalst, W.M.P., Vanwersch, R.J.B., Moleman, A.J.: Process mining in healthcare: Data challenges when answering frequently posed questions. In: Lenz, R., Miksch, S., Peleg, M., Reichert, M., Riaño, D., ten Teije, A. (eds.) ProHealth 2012/KR4HC 2012. LNCS, vol. 7738, pp. 140–153. Springer, Heidelberg (2013)
21. Mans, R.S., Schonenberg, H., Song, M., van der Aalst, W.M.P., Bakker, P.J.M.: Application of process mining in healthcare - a case study in a dutch hospital. In: BIOSTEC (Selected Papers), pp. 425–438 (2008)
22. Martinenghi, D., Torlone, R.: A logical approach to context-aware databases. In: D'Atri, A., De Marco, M., Braccini, A.M., Cabiddu, F. (eds.) Management of the Interconnected World, pp. 211–219. Physica-Verlag HD (2010)
23. Metzger, A., et al.: Disambiguating the documentation of variability in software product lines: A separation of concerns, formalization and automated analysis. In: RE, pp. 243–253 (2007)
24. Mori, M., Cleve, A.: Feature-based adaptation of database schemas. In: Botterweck, G. (ed.) MOMPES 2012. LNCS, vol. 7706, pp. 85–105. Springer, Heidelberg (2013)
25. Mori, M., Li, F., Dorn, C., Inverardi, P., Dustdar, S.: Leveraging state-based user preferences in context-aware reconfigurations for self-adaptive systems. In: Barthe, G., Pardo, A., Schneider, G. (eds.) SEFM 2011. LNCS, vol. 7041, pp. 286–301. Springer, Heidelberg (2011)
26. Nezhad, H.R.M., Saint-Paul, R., Casati, F., Benatallah, B.: Event correlation for process discovery from web service interaction logs. VLDB J. 20(3), 417–444 (2011)
27. Nzekwa, R., Rouvoy, R., Seinturier, L.: A flexible context stabilization approach for self-adaptive application. In: PerCom, pp. 7–12 (2010)
28. Parra, C., et al.: Using constraint-based optimization and variability to support continuous self-adaptation. In: SAC, pp. 486–491 (2012)
29. Parra, C., Cleve, A., Blanc, X., Duchien, L.: Feature-based composition of software architectures. In: Babar, M.A., Gorton, I. (eds.) ECSA 2010. LNCS, vol. 6285, pp. 230–245. Springer, Heidelberg (2010)
30. Quintarelli, E., Rabosio, E., Tanca, L.: Context schema evolution in context-aware data management. In: Jeusfeld, M., Delcambre, L., Ling, T.-W. (eds.) ER 2011. LNCS, vol. 6998, pp. 290–303. Springer, Heidelberg (2011)
31. Rebuge, Á., Ferreira, D.R.: Business process analysis in healthcare environments: A methodology based on process mining. Inf. Syst. 37(2), 99–116 (2012)
32. Roy, B.: Multicriteria Methodology for Decision Aiding. Kluwer Academic Publishers (1996)
33. Rozinat, A., de Jong, I.S.M., Günther, C.W., van der Aalst, W.M.P.: Process mining applied to the test process of wafer scanners in asml. IEEE Transactions on Systems, Man, and Cybernetics, Part C 39(4), 474–479 (2009)
34. Salehie, M., Tahvildari, L.: Self-adaptive software: Landscape and research challenges. TAAS 4(2) (2009)
35. Saval, G., Puissant, J.P., Heymans, P., Mens, T.: Some challenges of feature-based merging of class diagrams. In: VaMoS, pp. 127–136 (2009)
36. Schäler, M., Leich, T., Rosenmüller, M., Saake, G.: Building information system variants with tailored database schemas using features. In: Ralyté, J., Franch, X., Brinkkemper, S., Wrycza, S. (eds.) CAiSE 2012. LNCS, vol. 7328, pp. 597–612. Springer, Heidelberg (2012)
37. Schobbens, P.-Y., Heymans, P., Trigaux, J.-C., Bontemps, Y.: Generic semantics of feature diagrams. Computer Networks 51(2), 456–479 (2007)

38. Siegmund, N., Kästner, C., Rosenmüller, M., Heidenreich, F., Apel, S., Saake, G.: Bridging the gap between variability in client application and database schema. In: BTW (2009)
39. Terwilliger, J.F., Cleve, A., Curino, C.A.: How clean is your sandbox? In: Hu, Z., de Lara, J. (eds.) ICMT 2012. LNCS, vol. 7307, pp. 1–23. Springer, Heidelberg (2012)
40. van der Aalst, W.M.P.: Process mining: Overview and opportunities. ACM Trans. Management Inf. Syst. 3(2), 7 (2012)
41. van der Aalst, et al.: Process mining manifesto. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM Workshops 2011, Part I. Lecture Notes in Business Information Processing, vol. 99, pp. 169–194. Springer, Heidelberg (2012)
42. Villegas, A., Olivé, A.: A method for filtering large conceptual schemas. In: Parsons, J., Saeki, M., Shoval, P., Woo, C., Wand, Y. (eds.) ER 2010. LNCS, vol. 6412, pp. 247–260. Springer, Heidelberg (2010)
43. Villegas, A., Olivé, A., Sancho, M.-R.: On computing the importance of associations in large conceptual schemas. In: Düsterhöft, A., Klettke, M., Schewe, K.-D. (eds.) Conceptual Modelling and Its Theoretical Foundations. LNCS, vol. 7260, pp. 216–230. Springer, Heidelberg (2012)
44. Vincke, P.: Multicriteria Decision-Aid. J. Wiley, New York (1992)

Formalizing Service Variability Modeling in SOA-Based Solutions

Karthikeyan Ponnalagu[1,3], Nanjangud C. Narendra[2], and Aditya Ghose[3]

[1] IBM Research India, Bangalore, India
karthikeyan.ponnalagu@in.ibm.com
[2] IBM India Software Lab, Bangalore, India
ncnaren@gmail.com
[3] University of Wollongong, Austria
aditya.ghose@gmail.com

Abstract. Typical SOA-based solution design involves development of multiple inter-connected models using model-driven development (MDD) techniques. Hence these models are first created in platform neutral form and subsequently transformed through decreasing levels of abstraction before getting into executable form. Therefore creating and reusing variations of these models, for the purpose of enhancing reuse is a difficult challenge. In our earlier work, we had proposed techniques for developing variability models and deriving valid variants of services in a SOA-based solution. But our earlier work lacked a formal semantics for modeling and generalizing variations at different levels of abstraction. In this paper, we present the formal semantics via our *Variability Algebra*. Via this algebra, we show how variation oriented design of SOA-based solutions can be made a formalized, repeatable and verifiable exercise that helps maximize reuse. We also demonstrate theoretical results that can help optimize the generation and integration of service variants into an SOA-based solution. Throughout this paper, we illustrate our ideas on a running example.

Keywords: Variability, Reuse, SOA, Service Modeling.

1 Introduction

Service Oriented Architecture (SOA)-based solution development [14] involves design and development of business architecture models (such as process models or business component models), solution architecture models (such as service models or data models) and implementation specification models (such as service component architecture (SCA)[1] specification models). Model Driven Development (MDD) techniques, primarily considered for developing SOA-based solutions, enable business analysts and solution architects to better integrate these independently developed models [15]. One of the main objectives of developing SOA-based solutions is to maximize reuse of existing services [4]. It is well

[1] http://www.ibm.com/developerworks/library/specification/ws-sca/

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 402–416, 2013.

known that re-usability of any element such as a service or a business process is directly dependent on the extent of its variability, i.e., the extent to which it can be modified. This greatly influences in considering or rejecting an asset such as a SOA-based application in terms of its reduced customization cost against its initial acquisition cost [25]. There has been extensive work on variability management (e.g., [13,6]), where the overall reuse context is typically focused from the initial development perspective. In our earlier work [20] we proposed a methodology called "variation oriented engineering" for generating controlled variations of individual services and business processes from consumption perspectives. In [21] we addressed the derivation and selection of service variants to support the required changes in a business process specification model. In [19] we formalized the notion of variability model in a SOA-based solution. But this paper is motivated by several challenges that still exist in reusing SOA-based applications. First, the overall reuse context is currently dominated from the initial development perspective. This upfront and one time variability management may be acceptable in product line architectures, but not for SOA-based solutions, where subsequent reuse considerations could widely differ with changing business dimensions. In other words, mechanisms for formally determining all possible variations in a SOA-based solution, are lacking. Second, there is a lack of formal mechanisms to represent and help differentiate variants derived out of different reuse contexts from a SOA-based application over a period of time. This leads to development of redundant variations, that are already available from earlier reuse considerations on the application. Third, models belonging to different layers of the SOA stack (such as process and service models), are only interlinked with dependency relationships, but not formally integrated due to the limitations of current SOA modeling tools. This causes independently generated variations of these models to be inconsistent with each other and unusable together, thereby hindering effective reuse.

1.1 Key Contributions

To address the aforementioned challenges, in this paper we propose the *Variability Algebra* for SOA-based solutions. The variability algebra provides a formal framework that construct and leverage the lattice structures [9] for service variability modeling. The operators defined as part of the variability algebra, enable integrated variability modeling for services and processes in a SOA-based application. They also help generate complete solution variants based on consistent variation mechanisms. This extends the proposed techniques in [20], where we focused primarily on independent service and process variations. With the variability algebra, for each service or a process in a SOA service model, a lattice comprising a set of all its variants is constructed. Then for each service taking part in a process design, a notion of minimal dependency relation is evaluated. This leads to the mapping of the variants in the service lattice to the corresponding variants in the process lattice. Thus the operators enable a comparative validation of variants of different services taking part in a solution context for identifying any potential re-usability conflicts. Now, such an analysis for example, could ensure that all considered services contain non-conflicting variations, before engineering them into the derivation of respective process variants. In [21], we presented an algorithm to generate service variants from business process specifications without such a formal verification. The operators also enable analysis of existing variants, as well as modification thereof in response to changing requirements. This enables continuous validation of variation models [19] (in terms of applicable variation points, variations and allowed variants) as their associated constructs such as services or business processes evolve. Hence our proposed variability algebra intends to make variation-oriented design a formalized, repeatable and verifiable exercise that helps maximize reuse. From an ease of use and implementation perspective, the variability operators can be implemented and seamlessly integrated as plugins into MDD tools.

This paper is organized as follows. Section 2 introduces our running example, which we will be using throughout the rest of our paper for illustration. The basic notations and definitions of the constructs of our variability algebra are explained in Section 3. Related work is discussed in Section 4. Finally, the paper ends in Section 5 with suggestions for future work.

2 Running Example

Our running example comprises an insurance claims business process `Pr1` as illustrated in Fig. 1. The inputs to this process are the details of the customer requesting the claim, and the details of the claim. The outputs of this process are the acceptance/rejection of the claim, along with the claim amount to be paid to the customer (which will be zero in case of rejection). `Pr1` consists of four major sub-processes - (i) `Record Claim` (also referred to interchangeably henceforth as RC), (ii) `Verify Claim` (also referred to interchangeably henceforth as VC), (iii) `Analyze Claim` (also referred to interchangeably henceforth as AC) and (iv) `Report` (also referred to interchangeably henceforth as RP). In `Verify Claim` sub-process, the *DetermineLiability* (also referred to interchangeably henceforth as *DL*) and *PotentialFraudCheck* (also referred to interchangeably henceforth as *PF*) services are first executed in parallel, and then their results are combined and sent to *ClaimInvestigation (CI)* service. A final review of the verified claim is then implemented by *FinalReview (FR)* service. Now, let us assume another process `Pr2`, with the following differences from `Pr1`: *DetermineLiability* and *PotentialFraudCheck* services are executed in serial order; and *PotentialFraudCheck* service is modified to consider the output of *DetermineLiability* (i.e., the extent of liability now becomes a factor in checking for fraud). In the subsequent sections, we discuss how we can formalize and validate the derivation of `Pr2` in Fig. 2 from `Pr1`, based on the available and non-conflicting variations of the services `PF, DL` and the process `Pr1`.

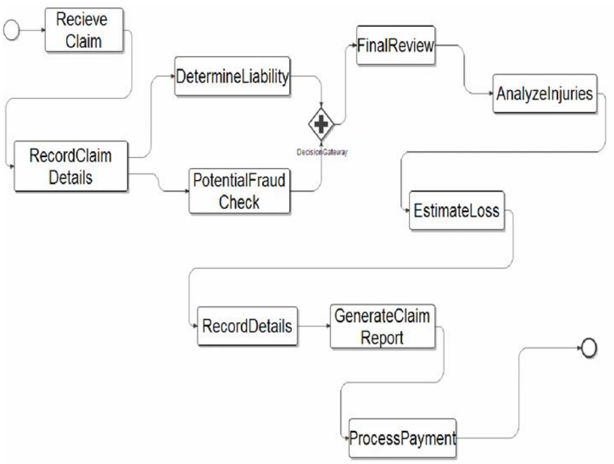


Fig. 1. Insurance Claims Process

3 Our Variability Algebra

We first define our domain of discourse, viz., our SOA-based solution model, as a set E. This set contains elements such as process model, service model, data model, etc., along with each of their variants. We represent these elements as a partial order with the binary inclusion relation $\leq$, as follows: for any two elements x and y, $x \in E$ and $y \in E$, $x \leq y$ if x is needed in order to develop y. For example, x and y would share the inclusion relationship so described if y was a variant of x, since x is needed in order to construct y. Another example would be if y were a process model and x a service model needed for building y. The primary use of this partial order is to integrate the independently developed variations of each element in an SOA-based model. For ease of exposition, this can be pictorially illustrated via a Hasse diagram [5] as in Fig. 3 for various variants of the `VC` sub-process. (Later in this paper, we show how it can be formally represented as a lattice [9], which helps optimize variant selection.)

For this paper, however, the scope of our variability algebra is limited to two primary elements of an SOA-based solution, viz., service and process models. Incorporating other elements (e.g., data models, component models) will be taken up in future work.

3.1 Definitions

In this section, we provide the formal semantic definitions for the basic SOA constructs and for the proposed variation constructs.

Definition 1 A *service operation* $Op = (in, out, b)$ is defined as a triple consisting of input data (in), output data (out) and implementation (b). ∎

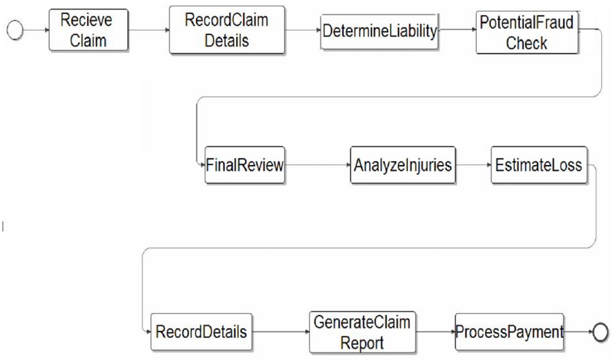


Fig. 2. Sub-process Verify Claim in Pr2 from the running example

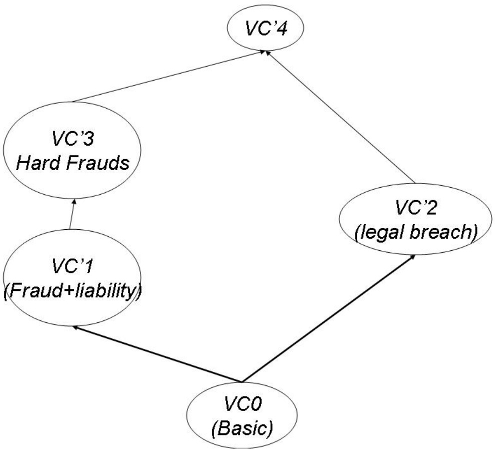


Fig. 3. Hasse Diagram of Verify Claim Sub-Process

Definition 2 A *service* $S = \{op_i \mid i = 1, 2...N\}$ is defined as a set of service operations; U is defined as the universe of all such services. ▮

Definition 3 A *variation point* VP_S of the service S is an item belonging to a triple Op of the service S belonging to U, the universe of all services. If the variation point is an implementation body (b) of Op, then it is an *implementation variation point*; otherwise, it is a *specification variation point*. S_{VP} is the set of all variation points associated with S. ▮

The declaration of the variation points need to be typically carried out by the service designers that designed the service. For instance, a non-declaration of the operation *fraudCheckAlgorithm()* in the service *PotentialFraudCheck* as a variation point, would invalidate the proposed derivation of the process `Pr2`.

Definition 4 A *variation* VF is defined as a function, which takes a service S and a variation point as input, and generates another service S' as output. It is formally expressed as $VF : U \times U_{VP} \longrightarrow U$. S_{VF} is the set of all variations associated with S. We also define this function so that the following properties hold:

- If the set of service operations of S and S' are OP_S and $OP_{S'}$, then $OP_{S'} \supseteq OP_S$
- For any service operation Op that belongs to both S and S', let the input and output sets be in_0, in_1, and out_0 and out_1, respectively. Then $in_1 \supseteq in_0$ and $out_1 \supseteq out_0$. If only the types of the input (resp. output) data are modified, then $in_1 = in_0$ (resp. $out_1 = out_0$). ▮

A variation for a service can be defined on a variation point either by the original designer of the service or subsequently by another designer that reuses the service with some controlled variations for a different application context.

Definition 5 A *variant* of a service S is defined as a service created by a composition of set of variations on S. A variant S' created with composition of n variations can be expressed as $S' = VF^n(\ldots VF^2(VF^1(S, VP^1_S), VP^2_S) \ldots VP^n_S)$ with the variation sequence : $VF^1, VF^2, \ldots VF^n$. U' is the set of all variants associated with S. ▮

Definitions 4 and 5 impose an inclusion relationship between a service and its variant, in that the variant is always constrained to “include” the base service. Please note, however, that this is not the same as the inheritance relationship [17] prevalent in object-oriented design; a variant of a service may not be able to “substitute” for its base service, since it may require more inputs than the base service. Conversely, the base service may also not “substitute” for its variant, since it may provide less outputs than its variant. A variant is thus basically created from the base service through a controlled set of variations validated by the declaration of variation points on the service. For example, let a variant S_1 be generated from S_0 by the addition of two input data d_1 and d_2, and let an additional variant S_2 be desired with only d_1 as the desired extra input. Then S_2 can only be generated directly from S_0 by adding the desired input d_1; S_2 *cannot* be generated from S_1 by removing d_2, since that would violate Definition 4.

The sequential composition approach helps in prioritizing the variations and also to resolve conflicts arising out of applying variations that can be mutually exclusive in nature. Also each variation function would only generate a modified service, if the specified variation point is still applicable on the service specified as input. As illustrated in Fig. 2, the derivation of variants achieved via allowed composition of variations can be mandated. In the running example, we observe the derivation of a variant for the sub process `Verify Claim` is allowed only from both the variations of the services `DetermineLiability` and `FraudCheck` respectively and not otherwise. Such a constraint can be relaxed in the future, when new variations are allowed for these services based on how they extend the respective base services.

Definition 6 A *Variation Model* $VM_S = (S_{VP}, S_{VF}, U')$ of a service S is a triple consisting of the following elements:

- The set of all variation points
- The set of all variations
- The set of all variants ∎

The variation model of a service is explicitly defined and non-intrusively associated with the corresponding service as a separate element. Thus it does not affect the core design constructs of the service that form part of a larger SOA-based solution. In our running example, the service *DetermineLiability* contains the service operation *liabilityCheckInfo()*, which is required for realizing the service *DetermineLiability* in `Pr1`. Since `Pr2` requires a fundamental change to *DetermineLiability*, this mandates that *DetermineLiability* needs to be first declared to support variation. Subsequently, the actual derivation of `Pr2` from `Pr1`, would only be possible, if the required variation is defined in the variation model for *DetermineLiability*. Also, the process `Pr1` itself needs to be declared as a variation point - this is possible by defining `Pr1` itself as a (composite) service. Such a declaration is mandatory for enabling the variant derivation of `Pr2` by changing the process flow.

The service operation *fraudCheckAlgorithm()* of the service *PotentialFraudCheck*, defines both specification and implementation variation types. The service *DetermineLiability* contains the operation *liabilityCheckInfo()*, which defines only a specification variation. For deriving the process `Pr2` from `Pr1`, the operation *fraudCheckAlgorithm()* in the service *PotentialFraudCheck* needs to contain just an implementation variation, since `Pr2` does not mandate any interface level changes in *fraudCheckAlgorithm()*. But in the case of *DetermineLiability* containing the operation *liabilityCheckInfo()*, we need to define a specification variation, since both the interface specification and corresponding implementation need to be modified for providing the required output for the service *FinalReview*. Thus subsequent to the definition of variation points across the different constructs of `Pr1`, the variation points need to be associated with corresponding types of compatible variation specifications for deriving not just `Pr2` as in the case of our running example, but also for other process or service variants in the future.

3.2 Properties of Our Variability Algebra

Our variability algebra is a category [28], as per the following theorem.

Theorem 1. For a defined Variability Model, the services and their variants that comprise our Variability Algebra form a category.

Proof: A category is a set of objects with morphisms (mappings) between them. In our case, the services and their variants are the objects and the variation functions are the morphisms. It obeys the following properties:

– Property $C0$: The set of services and their variants forms the objects of the algebra, and the variation functions linking services and their variants form the morphisms among the objects. Among two services that are not variants of each other, due to the absence of a variation function, a default null morphism can be defined.
– *Property C1 - Identity Element*: The trivial identity isomorphism, which maps a service onto itself, can be defined.
– *Property C2 - Associativity*: Morphisms are associative, i.e., if there exist variation functions $S_0 \xrightarrow{F_a} S_1$, $S_1 \xrightarrow{F_b} S_2$, and $S_2 \xrightarrow{F_c} S_3$, then $(F_a \circ F_b) \circ F_c = F_a \circ (F_b \circ F_c)$. That is, $F_a \circ F_b$ represents the (eventual) derivation of S_2 from S_0, while $F_b \circ F_c$ represents the (eventual) derivation of S_3 from S_1. Hence both $(F_a \circ F_b) \circ F_c$ and $F_a \circ (F_b \circ F_c)$ represent the (eventual) derivation of S_3 from S_0.
– *Property C3 - Pairwise Disjoint*: The morphisms are pairwise disjoint. That is, if there are two variation functions $S_0 \xrightarrow{F_a} S_1$ and $S_0 \xrightarrow{F_b} S_1$, with the same domain and range, then they are identical, since they produce the identical variant on the same base service. ∎

In addition to the properties proved in Theorem 1, the following property can also be defined.

Property C4: The variation function from Definition 4 that produces a variant S' from S is an isomorphism, since it is reversible, i.e. the reverse variation function that brings back S from S' is also possible. This is crucial for reversing any variation decisions during solution development. It is also easy to show that any composition of two such isomorphisms is also an isomorphism, thus ensuring that any sequence of variation functions can also be reversed.

Of course, Property $C4$ holds under the assumption that the variation model of the service remains unchanged during variant generation. For example, if the generation of S' from S were to involve the deletion of an internal operation in S, the reverse variant generation from S' to S would be to bring back the operation. In the meantime, if the operation in question itself would be removed from the variation model (i.e., no longer regarded as a variation point), then the reverse variant generation from S' to S would not be possible.

However, despite Property $C3$, it is to be noted that variation functions need not commute. Indeed, $F_a \circ F_b$ would require S_0 as its domain and S_2 would be its range. Whereas, F_b would require S_1 as its domain and S_2 as its range, and F_a would require S_0 as its domain and S_2 as its range. Therefore, $F_b \circ F_a$ would not be possible, since variant derivation implicitly implies a direction, from a base service to its variant.

Theorem 1 also raises the question of how to determine the correct variation function in order to derive a required variant S' from its base service S. Alternately, given a variation function represented as a set of variation actions on S, we need to formally verify whether it produces the variant S' that we need.

We now extend the above category-theoretic ideas to consider the composition of services in a business process as part of an SOA-based solution. But first we note that, since, as per Definition 4, a base service is always included in its variant, this inclusion relationship can be used to prove the following result.

Theorem 2. The base service and its variants form a poset [9].
Proof: Let the base service and its variants form a set P under the variant relationship as defined in Definition 4, and which we denote by $\lhd$. That is, $S_0 \lhd S_1$ if S_1 is a variant of S_0. Then the following properties of a poset can be proved directly from Definition 4:
Reflexivity: $S_0 \lhd S_0$
Antisymmetry: If $S_0 \lhd S_1$ and $S_1 \lhd S_0$, then $S_0 = S_1$
Transitivity: If $S_0 \lhd S_1$ and $S_1 \lhd S_2$, then $S_0 \lhd S_2$ ▮

From Theorem 2 the following theorem can be derived.

Theorem 3. The set P of base service and its variants form a lattice [9].
Proof: A lattice is a poset with the following additional properties:

- For any two elements S_i and S_j in the poset, we can always establish that there exists a join (least upper bound). If S_j is a variant of S_i, then the join is the variant itself. If S_i and S_j are variants of other base services, then the join is either a variant derived from S_i or S_j or the universal set containing all the services in P.
- For S_i and S_j, we can always establish that there exists a meet (greatest lower bound). If S_j is a variant of S_i, then the meet is S_i itself. If S_i and S_j are variants of other base services, then the meet is either a service from which S_i and S_j are derived, or the base service of P itself. ▮

From Theorem 3, we are able to develop useful results regarding the mappings between service variant lattices. But first we provide some definitions.

Definition 7 S_j has a *dataflow dependency* on S_i if $O_{S_i} \cap I_{S_j}$ is non-empty. That is, some outputs from S_i would be needed in order for S_i to execute. We formally represent this as $S_i \xrightarrow{d} S_j$. ▮

Definition 8 We define service lattices L and K as *composable* if the following holds for the base services A_0 and B_0 of L and K, respectively: $A_0 \xrightarrow{d} B_0$. ▮

Definition 9 Given two composable service lattices L and K, let us assume the existence of the following dataflow dependency between any two variants A_i and B_j of L and K, respectively: $A_i \xrightarrow{d} B_j$. Then this dependency is a *maximal* dependency if for any variant A_p of A_i, the following dependency does *not* hold: $A_p \xrightarrow{d} B_j$. ▮

Hence Definition 9 provides a means to select the so-called "best fit" variant A_i needed to compose with B_j in the solution. This leads us to the following result.

Theorem 4. Let a maximal dependency relation $A_i \stackrel{d}{\rightarrow} B_j$ exist between variants A_i and B_j in lattices L and K. Let B_m be a variant of B_j. If a maximal dependency relation $A_j \stackrel{d}{\rightarrow} B_m$ exists between B_m and a service A_j belonging to lattice L, then A_j is either A_i itself, or a variant of A_i.

Proof: There are three sub-cases:

– B_m does not contain any extra input data. Then the maximal dependency relation $A_i \stackrel{d}{\rightarrow} B_m$ holds.
– B_m contains extra input data, but which cannot be provided by any service in the lattice L. Here also, the maximal dependency relation $A_i \stackrel{d}{\rightarrow} B_m$ holds.
– B_m contains extra input data, which can be provided by one or more services in the lattice L. Then from Definitions 4 and 9, it follows that the only service A_j for which $A_j \stackrel{d}{\rightarrow} B_m$ holds, is a variant of A_i. ▮

A brief illustration of Theorem 4 is depicted in Fig. 4, which shows two service variant lattices of the services `DetermineLiability` and `PotentialFraudCheck` from our running example. For the original process `Pr1`, we know that $DL0$ is composable with $PF0$. For the variant process `Pr2`, we replace $DL0$ with its variant $DL'1$ to support sequential composition. As per Theorem 4 the only possible variants of `PotentialFraudCheck` lattice that are candidates for composition with $DL'1$ are $PF'1$ and $PF'5$. Furthermore, for `Pr2`, since we also require the extent of liability to determine fraud, the selected variant of `PotentialFraudCheck` is $PF'5$. In other words, Theorem 4 shows that the maximal dependency relation is an order-preserving morphism between the two

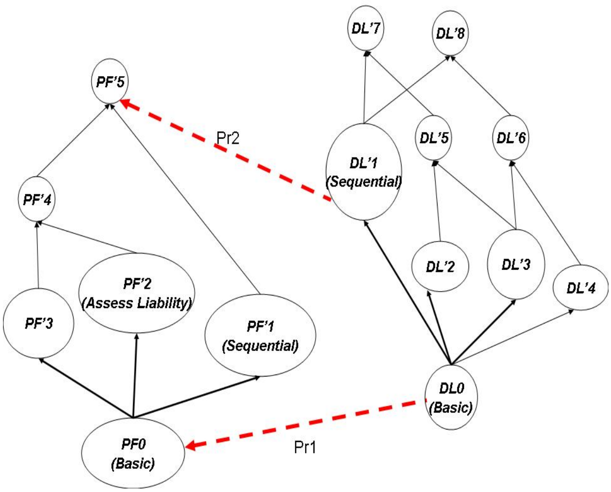


Fig. 4. Illustration of Theorem 4

service lattices. That is, assuming a maximal dataflow dependency among the services in an SOA-based solution, let us assume that an existing service S_j is replaced by its variant. Then for another service S_i on which S_j is dependent, we need only search among the variants of S_i to determine the appropriate variant of S_i to compose with that of S_j. This greatly eases the task of selecting the appropriate service variants to generate a required variant for an SOA-based solution.

From an evaluation perspective, we can use category-theoretic ideas (Theorem 1) to help us define and generate variants and maintain the variation model of a service. Once this is done, we can use lattice-theoretic ideas (Theorems 2, 3 and 4) to optimize the integration of generated variants into a business process as part of an SOA-based solution. This is achieved with establishing the minimal dependency relations between the variants of a service lattice with the variants of the other service lattices required in the business process. This would therefore help automate variation-oriented design to a large extent, thereby rendering it potentially scalable on large real-life systems.

3.3 Operators of our Variability Algebra

One of the primary requirements of any algebra is a set of operators that be applied on the constructs belonging to the algebra. From the viewpoint of SOA-based reuse, the operators that we define need to help in automating the key steps of a variation-oriented design approach, viz., creating variation model, generating a service variant, and validating the generated variant. To that end, we now define and discuss a core set of variability operators. The operators can be implemented independently and integrated in a service Modeling tool such as [3].As these operators are modular in nature, they can be independently used during the design of a SOA-based application by the architects and designers.

- **CreateVariationModel(S)**: This operator creates (and updates) the variation model of the service S. This operator is used when variation points, variation actions are added or deleted, and when new variants are generated. In our running example, for the service *PF*, its variation model would consist of the following:
  - variation points: the service *PF* itself and its operation *fraudCheckAlgorithm()*;
  - variation actions: *fraudCheckAlgorithm()* modification that considers *liabilityInfo* as additional input and *fraudCheckAlgorithm()* modification that enables the operation to be invoked sequentially; and
  - variants: *PF'1, PF'2, PF'3, PF'4, PF'5.*
- **CreateVariantLattice(S):** This operator creates (and updates) the variant lattice of the service S as defined by leveraging the inclusion relationships established between the service and its variants and is illustrated in Fig. 3 for the *VC* sub-process. This operator depends on the existence of the variation model for a given service.

- **GenerateVariant**$(S, \{VF^i\})$**:** This operator generates a variant S' of S by applying the variations VF^i as per Definition 5 and Property C0 in Theorem 1. Examples are the generation of the variants of *DetermineLiability* and *PotentialFraudCheck* in our running example. First the service designer selects the list of available variations from the variation model and then invokes this operator to generate the resulting variants. These are checked for uniqueness against already generated variants via Propery $C3$ from Theorem 1.
- **ValidateVariant**(S, S')**:** This operator validates the correctness of a variant S' of S in terms of the current variation model of the service S. This is needed for continuous evaluation of an existing variant against changing variation model of the corresponding base service. A variation model can be subjected to changes such as removal or addition of variation points or variations due to domain related constraints. From Definition 5, we can express S' as $S' = VF^n(\ldots VF^2(VF^1(S, VP^1_S), VP^2_S)\ldots VP^n_S)$. Hence S' is a valid variant of S, only if the following holds true: $\{VF^1, VF^2, \ldots VF^n\} \subset S_{VF}$. This subsequently mandates that $\{VP^1, VP^2, \ldots VP^n\} \subset S_{VP}$. This means that there exists no variation action that has been used in the construction of S', but not defined as part of the set of variations defined for the service S. Now, if we assume $VP^1 \subset S_{VP}$ as not true. Then S' becomes invalid as a variant of S, as VF^1 can not be a supported variation in S_{VF}. From Theorem 3, we can now establish that S' is the join and S is the meet in $\{S, S'\}$.
- **PlaceVariant**(S'_x, E)**:** This operator is useful in placing the service variant S'_x in the lattice of a service S. This operator can be invoked post validating the variant using the previously discussed operator. The objective is to ensure that only unique and valid variants can be accepted into the variation model. The uniqueness is ensured from Definition 5 and Property $C2$.
- **IntegrateVariant**(S', P)**:** This operator integrates the selected variant S' belonging to the variant lattice whose base service is S, into the business process P in conformance with Theorem 4. This is illustrated in Fig. 4. This helps in assessing whether a newly derived variant S' of S helps in realizing one of the desired variants of process P listed in its lattice. The operator starts with the basic comparison of the variations S'_{VF} used to derive S' from S, with the defined variations P_{VF} of the process P. If $S'_{VF} \nsubseteq P_{VF}$, then the variant S' is rejected as there is a re-usability conflict with S'. Otherwise, for each variant P' of the process P, where $S'_{VF} \subseteq P'_{VF}$ is true, S' is integrated into P'.

4 Related Work

Most existing works in the area of software variability management have concerned themselves with variability management in product line architectures [31,30] and feature engineering [7,13,6,18]. On leveraging feature modeling for service variability [32,16,2], the underlying basic assumption is that the software developer is aware of the minimum set of features required in a service to fulfill a given functionality and that the service variant should possess the same set of minimum required inputs and outputs as the original service. In COVAMOF [10], variations are modeled from a product configuration and composition perspective. While such a constructive approach is very efficient, it deals with variations mainly in terms of adding new components or removing existing components and (re)setting parameters purely from a product line architecture perspective. Our work can complement such an approach for better organization and consistent repeatable derivation of variants of existing components. In related work on process flexibility and process variants [26,27], the authors discuss both process schema related and instance related changes from a change management perspective for processes at run time.In an earlier paper [23], we also investigated in greater depth the issue of maintaining semantic consistency among the design artifacts of a constantly evolving SOA-based solution via impact analysis. Our proposed variability algebra can therefore help in enforcing change related constraints formally during process design, as we define a boundary in terms of variation points and corresponding variation actions.

Fiammante et al. [11] discussed the categorization of SOA variability in terms of information variability, service variability and process variability and also the mechanisms involved in achieving these independently. Using our proposed variability algebra, we believe that such independently developed variations belonging to different types such as options, variants and boolean guards (as described in [31]) can be validated, integrated and represented in a single variation model to generate solution level variations. In our work, we also treat all variations as optional that can only be applied on the base elements such as services if needed. This is consistent with the notion of SOA unlike variability management in product line architectures, where unwarranted variations need to be explicitly declared optional to avoid any conflicts. Thus in contrast to [31], our work can enable different development teams to develop or reuse the required variations that are valid in terms of the solution's integrated variability model.

Puhlmann et al. [29] primarily concentrate on detailing variations expressed via the object-oriented concept of inheritance (i.e., interfaces are considered to be invariant). In contrast, our approach follows an explicit notion of variability modeling for interconnected models. While [6] provides a taxonomy of specific variability types (workflow, composition, interface and logic), our work considers and implements the actual semantics of variations. The variation relations between the generated variants discussed in [10] can easily be represented and cross validated through the lattice based formalization we defined as part of our variability algebra. As service substitutability in [8], the authors discuss how a service can be replaced with another service through domain ontological matching. This basically assumes the services to have the same functionality, but with different syntactic or semantic definitions. Our work complements [8] in deriving such functional variants. Ralyte et al. [24] propose operators on integrating scenarios based on process or product level contexts. We see our work extending this notion of integration further in terms of deriving variants that can actually help realize these scenarios.

5 Conclusions

In this paper, we have addressed the issue of turning variation oriented design of SOA-based solutions into a formalized, repeatable and verifiable exercise. To that end, we identified the key challenges that are still unresolved despite extensive work so far in variability management, In order to address these challenges, we presented our *variability algebra*, which aims to formalize variant generation and ensure its correctness in order to facilitate automation. We have also illustrated our algebra on a running example. For future work, we will be investigating implementation of our algebra via automated business process composition using lattice homomorphisms between service variant lattices. In particular, our investigations will focus on automated cross-verification techniques [12] for verifying the correctness of generated variants that can be selected for integrating into SOA-based solutions as per Theorem 4. For this, we will be enhancing our earlier service variant derivation algorithm from [21] using the properties of our variability algebra. We will also be investigating the approach presented in [1] for automated variability artifact composition to automatically compose service variants to develop business process variants. We also plan to leverage [22] for comparing and drawing conceptual similarity between independently developed variants. This would help in representing and reusing variations (which may not be structurally same but semantically equivalent), thereby further enhancing reuse. We will also be investigating the inclusion of contextual parameters such as standards compliance during variant derivation and selection for composition.

References

1. Acher, M., Collet, P., Gaignard, A., Lahire, P., Montagnat, J., France, R.: Composing multiple variability artifacts to assemble coherent workflows. Software Quality Journal 20, 689–734 (2012)
2. Alferez, G.H., Pelechano, V.: Systematic reuse of web services through software product line engineering. In: Web Services (ECOWS), pp. 192–199 (September 2011)
3. Arsanjani, A.: Soma-me: A platform for the model-driven design of soa solutions. IBM Systems Journal 47(3), 397–414 (2008)
4. Bianchini, D., Cappiello, C., De Antonellis, V., Pernici, B.: P2S: A methodology to enable inter-organizational process design through web services. In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) CAiSE 2009. LNCS, vol. 5565, pp. 334–348. Springer, Heidelberg (2009)
5. Braggemann, R., Patil, G.P.: Partial order and hasse diagrams 5, 13–23 (2011)
6. Chang, S.H., Kim, S.D.: A variability modeling method for adaptable services in service-oriented computing. In: SPLC, pp. 261–268 (2007)
7. Czarnecki, K., Wasowski, A.: Feature diagrams and logics: There and back again. In: SPLC, pp. 23–34 (2007)
8. Valeria, D., Michele, M., Barbara, P., Pierluigi, P.I.: A methodology for e -service substitutability in a virtual district environment. In: Eder, J., Missikoff, M. (eds.) CAiSE 2003. LNCS, vol. 2681, pp. 552–567. Springer, Heidelberg (2003)
9. Davey, B.A., Priestly, H.A.: Introduction to Lattices and Order. Cambridge University Press (1990)
10. Deelstra, S., Sinnema, M., Bosch, J.: Variability assessment in software product families. Information & Software Technology 51, 195–218 (2009)
11. Fiammante, M.: Dynamic SOA and BPM: Best Practices for Business Process Management and SOA Agility, 1st edn. IBM Press (2009)
12. Ghezzi, C., Jazayeri, M., Mandrioli, D.: Fundamentals of Software Engineering, 2nd edn. Prentice Hall PTR, Upper Saddle River (2002)

13. Gomaa, H., Olimpiew, E.M.: Managing variability in reusable requirement models for software product lines. In: Mei, H. (ed.) ICSR 2008. LNCS, vol. 5030, pp. 182–185. Springer, Heidelberg (2008)
14. Huhns, M.N., Singh, M.P.: Service-oriented computing: Key concepts and principles. IEEE Internet Computing 9(1), 75–81 (2005)
15. Johnson, S.K., Brown, A.W.: A model-driven development approach to creating service-oriented solutions. In: Dan, A., Lamersdorf, W. (eds.) ICSOC 2006. LNCS, vol. 4294, pp. 624–636. Springer, Heidelberg (2006)
16. Lee, J., Kotonya, G.: Combining service-orientation with product line engineering. IEEE Software 27(3), 35–41 (2010)
17. Liskov, B.H., Wing, J.M.: A behavioral notion of subtyping. ACM Transactions on Programming Languages and Systems 16, 1811–1841 (1994)
18. Loesch, F., Ploedereder, E.: Restructuring variability in software product lines using concept analysis of product configurations. In: Conference on Software Maintenance and Reengineering, pp. 159–170 (2007)
19. Nanjangud, C.N., Karthikeyan, P.: Towards a variability model for soa-based solutions. In: IEEE SCC, pp. 562–569 (2010)
20. Nanjangud, C.N., Karthikeyan, P., Biplav, S., Guruduth, S.B.: Variation-oriented engineering (voe): Enhancing reusability of soa-based solutions. In: IEEE SCC (1), pp. 257–264 (2008)
21. Ponnalagu, K., Narendra, N.C.: Discovering and deriving service variants from business process specifications. In: Bouguettaya, A., Krueger, I., Margaria, T. (eds.) ICSOC 2008. LNCS, vol. 5364, pp. 691–707. Springer, Heidelberg (2008)
22. Akkiraju, R., Farrell, J., Miller, J., Nagarajan, M., Schmidt, M., Sheth, A., Verma, K.: Web service semantics: Wsdl-s. w3c Member (2005)
23. Ramya, R., Nanjangud, C.N., Karthikeyan, P., Dipayan, G.: Morpheus: Semantics-based incremental change propagation in soa-based solutions. In: IEEE SCC, pp. 193–201 (2008)
24. Ralyté, J., Rolland, C., Plihon, V.: Method enhancement with scenario based techniques. In: Jarke, M., Oberweis, A. (eds.) CAiSE 1999. LNCS, vol. 1626, pp. 103–118. Springer, Heidelberg (1999)
25. Ravichandran, T., Marcus, A.: Rothenberger. Software reuse strategies and component markets. Commun. ACM 46(8), 109–114 (2003)
26. Reichert, M., Dadam, P., Rinderle-Ma, S., Jurisch, M., Kreher, U., Göser, K.: Architectural principles and components of adaptive process management technology. In: PRIMIUM, pp. 81–97 (2009)
27. Reichert, M., Bauer, T.: Supporting ad-hoc changes in distributed workflow management systems. In: Meersman, R., Tari, Z. (eds.) OTM 2007, Part I. LNCS, vol. 4803, pp. 150–168. Springer, Heidelberg (2007)
28. Rotman, J.J.: Advanced Modern Algebra (Graduate Studies in Mathematics), 2nd edn. American Mathematical Society (2010)
29. Schnieders, A., Puhlmann, F.: Variability mechanisms in e-business process families. In: BIS, pp. 583–601 (2006)
30. Sinnema, M., Deelstra, S.: Industrial validation of covamof. Journal of Systems and Software 81(4), 584–600 (2008)
31. van der Hoek, A.: Design-time product line architectures for any-time variability. Sci. Comput. Program. 53(3), 285–304 (2004)
32. Wittern, E., Zirpins, C.: On the use of feature models for service design: The case of value representation. In: Cezon, M., Wolfsthal, Y. (eds.) ServiceWave 2010 Workshops. LNCS, vol. 6569, pp. 110–118. Springer, Heidelberg (2011)

Towards Understanding Software Process Variability from Contextual Evidence of Change

Tomás Martínez-Ruiz, Francisco Ruiz, and Mario Piattini

Alarcos Research Group, Information Technologies and Systems Institute,
University of Castilla-La Mancha
Paseo de la Universidad, 4, 13071 Ciudad Real, Spain
tomas.mrtnez@gmail.com, {francisco.ruizg,mario.piattini}@uclm.es

Abstract. Software development enterprises need to tailor their own processes before enact them in order to ensure that they fit both the organization and the project. This necessity has, to date, been solved by providing these processes with variability support. Tailoring proposals have traditionally been focused on solving the problem of managing the variability of processes in order to facilitate their adaptation. Process tailoring has not, however, been considered as a solution to a wider problem consisting of the organization, project, laws and some other influencing factors that change according to each project, a problem that software processes must confront if they are to be successful. In this paper we enhance a tailoring framework in order to tackle changes in the context level of the process, and this variability is considered to drive the tailoring of the supporting processes. As a part of the enhanced framework, this paper analyzes the OMG's Business Motivation Model (BMM) in order to apply it to the characterization of the organizational units as a part of the context variation factors, and to link them with subsequent process variations. The proposal is illustrated by means of an application example, which is based on a real industrial case and which has served as a proof of concept. The resulting conclusion is that since software process tailoring depends on the process context, so understanding and managing changes in the latter's drive variability in software processes.

Keywords: context change, evidence in software process, Variant-Rich Processes, tailoring management, process institutionalization, project management.

1 Introduction

The importance of process tailoring is no longer called in question. Software development organizations know that if they wish their projects to be successful, then they must use processes that meet the reality in which they are involved [1]. Tailoring is currently addressed by means of process variability techniques, which are in most cases applied from software products to software processes [2, 3]. Literature provides proof of this, and therefore includes several approaches, such as those proposed by Simidchieva et al [4] or Martínez-Ruiz et al [5].

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 417–431, 2013.

Tailoring has traditionally been treated as a problem that can be solved by means of variability support and other approaches, such as method engineering [6]. Following this line, in previous works we have presented both the Variant Rich Process paradigm [7] and the SPRINTT environment [8]. These are focused on providing process variability support, which is used in process institutionalization (embedding processes inside the organization through its tailoring from the self set of standard processes). They have been tested by using case studies and experiments, and although the results are positive, variability has been focused on the processes themselves, and the reflection of the similarities and differences to the organizations and projects in which these processes are systematically put into operation has been not considered.

In this paper, the process tailoring problem is placed in the context of a wider and more profound problem: *organizations, work teams, projects and even laws,* which constitute the *software process enactment context,* change. Since these have an impact on the software development, the processes must be tailored in order to engrain them and to include these changes. From an overall organizational viewpoint, alignment and traceability between the elements of which the process contexts and the processes themselves are composed must therefore be clearly determined, as must how the evidence of changes in the former affect the latter.

The proposed approach aims to embed the process variability and tailoring mechanisms, even the institutionalization ones, into a large environment, which also has the capability of meshing these mechanisms with the context changes. The proposal, called SPICCE (Software Process Institutionalization based on Context Change Evidence), considers that process variations may and must be developed from variations in the context, which clearly ensures their alignment and traceability with organizational strategies and project necessities. It is consequently promoted in order to apply the tailoring philosophy, i.e., to avoid creating a new process for each enactment to the process tailoring itself, by avoiding the need to adapt entire processes from scratch, and is instead driven by the changes in the context, which determine how to modify the adaptation itself from a process to another, to take into account the specific context requirements of the new one.

The research presented herein characterizes the distinct elements that a context is composed of: an organization's features, its motivation and rules, the projects and the external laws. As first step, it focuses on analyze the organization features, and proposes the elicitation of the changes in business by using feature models, which have been aligned with the OMG's Business Motivation Model (BMM) [9]. It also shows how to trace these high abstraction changes in concrete variations, which we propose to automate in future works. It opens the door to homogenize BMM with other business norms and standards, as well as the regulations that influence process enactment, and which change in between these processes.

In addition to this introduction, this article includes an overview of the state-of-the-art in Section 2, which reviews the process tailoring initiatives found in literature, along with a description of the elements included at organizational levels. The third Section presents the SPICCE proposal and the elements of which it is composed, and also includes an analysis of how BMM could be enhanced in order to make the changes in software organizations explicit. Finally, our conclusions and future work are described in Section 4.

2 State of the Art

Literature includes several initiatives concerning process tailoring, which are in most cases supported by means of modifications in process structures [10, 11]. Rombach then developed the "process lines" approach [2], while Sutton developed the links between aspects and processes [3]. The systematic literature review Martínez Ruiz *et al.* [12] presented describes how processes are modified in order for them to fit the project's needs; it also states the set of requirements a process variability notation must include to support tailoring as real organizations actually need it. It therefore guides the definition of new process variability support mechanisms.

Some other new work has also appeared since the aforementioned systematic review. Simidchieva *et al.* [4] present an explicit differentiation between problem and solution spaces, and identify three types of approaches: generation, navigation and reasoning. Araujo *et al.* [13] propose the management of process variability by identifying the common and different features of a process model. They propose a tool with which to apply MDE transformations to process tailoring [13]. Hurtado Alegría *et al.* [14] also propose tailoring software processes by using MDE and ATL transformations. Simmonds *et al.* [15] propose the creation of Basic Feature Models to represent features of the tailored process in the vSPEM notation, which constitutes our previous contribution (see section 3.1), paying special attention to orthogonal variations. These "new" approaches do not meet at all the requirements that industries require, as has been pointed out in the aforementioned SLR [12].

Of those initiatives which apply variability to software processes in order to align the processes themselves with the projects found, we should highlight the work of Martins and Silva [16, 17], a proposal based on three fundamental steps: i) defining the process, ii) adapting and monitoring the process execution, and iii) measuring the process. Killisperger *et al.* [18] suggest an environment with which to automatically apply variations to processes through the use of variation operations. Silva Barreto *et al.* [19, 20] propose another environment in which to carry out variations in software processes, with the aim of facilitating process reuse, based on the definition of variations in process components. These proposals include some aspects that seek process institutionalization, but none of them really align it with process tailoring.

With regard to tailoring and variability in the description of the organization, there are several works concerning variability in business processes, which also apply *Product Lines* or *Aspects*. Some of these works are those of Lu and Sadiq [21-23], or the AO4BPEL project of *Mezini et al.* [24-26], and they are even considered as families [27]. However, these works are not focused on discovering how the variations in the business processes affect their organization's software processes.

On the contrary, there are a few works about how the context influences business processes, but they show it is a relevant topic in the case of business processes. Ploesser *et al.* [28] presents the necessity of tailoring business processes in accordance with the context and identifies the techniques addressing it. Rosemann *et al.* [29] propose an onion layer system in which classify environmental, external, internal, and immediate influencers. In case of Product Lines, relationships between the context and the technical support have been addressed by using feature models to represent changes even in the context, as Hartman *et al.* proposes [30].

Bearing all of the above in mind, in addition to our goal of linking software process variations with the real facts motivating them in the process context, it is necessary to enhance the previously existing SPRINTT approach through the use of context change evidence management.

3 Process Institutionalization Based on Context Change Evidence

3.1 Previous Works: SPRINTT and VRP

The *Software Process Institutionalization based on Tailoring and sTandardization* (SPRINTT) approach is composed of two elements, which have been termed as the *Institutionalization Cycle* and the *Variant-Rich Process paradigm* (VRP) [7, 31, 32]. The former is the theoretical setting used by organizations to transform and include processes as new and effective assets. It defines four cyclical steps with which to *tailor*, *execute*, *analyze* and *standardize* the processes, as **Fig. 1** shows.

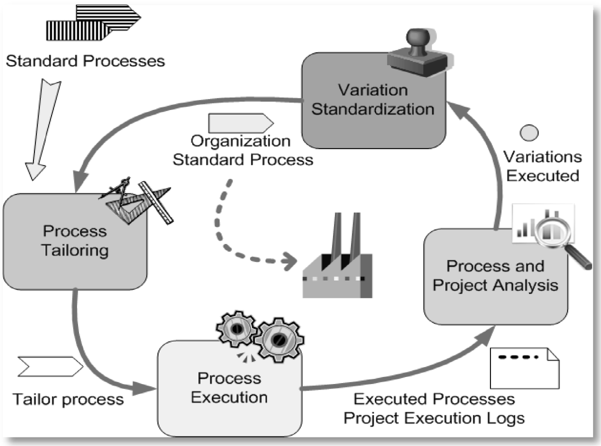


Fig. 1. Institutionalization Cycle overview

The Variant-Rich Process paradigm offers variability support which is suitable for the execution of both the first and last steps of the cycle. The paradigm provides on-point and crosscutting variability mechanisms, in addition to Rationale to support variation decisions. These are based, respectively, on Product Lines (SPLE's) [33], Aspects (AOSE) [34] and Rationale Management [35] from Software Engineering.

The VRP includes generic variability mechanisms. These have been implemented in SPEM [36], resulting in the vSPEM language [5, 7, 37-40], which provides tailoring support, as industries require. The use of the vSPEM notation makes it possible to manage on-point and crosscutting variations in software processes.

The VRP paradigm has been tested using the vSPEM notation. Experiments have shown that it is much easier to adapt processes if they are modeled using the vSPEM language [7]. These results have also clearly demonstrated that the VRP and the notation are useful in modeling variability in Aerospatiale, Quality Assurance, and Global Software Development domains [5, 31, 38].

This framework is focused on solving the problems that the industrial and scientific communities were worried about several years ago, when our research into this area began, as these communities were totally focused on managing variability in software processes as a reflection of the product's variability support in product lines.

3.2 Enhancing SPRINTT through Seeking Evidence: SPICCE

SPRINTT was designed and built, to provide a solution to process tailoring. The VRP takes a set of tailoring requirements and, by using rationale mechanisms, finds which is the best variation supporting each requirement, which is implemented by the variability mechanisms. **Fig. 2** presents an overview of this tailoring approach.

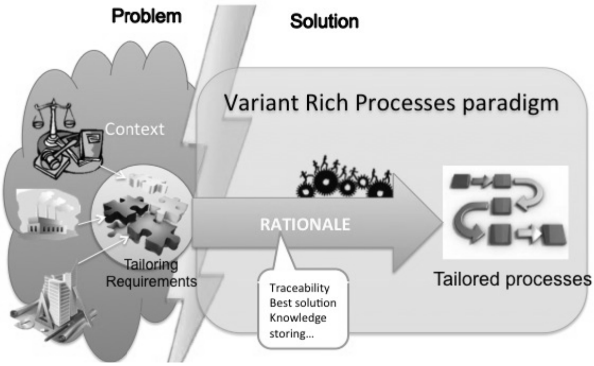


Fig. 2. Tailoring following the Variant Rich Process Paradigm

As a consequence of considering the feedback of the previous validations, we now know that a set of requirements is not a randomized creation. They are generated as a conjunction of three main elements surrounding the process enactment: the *project* (and its characteristics), the *organization* (although the entity focused on in software development is sometimes not the organization, but a department), and the *laws or regulations* (affecting the organization), as left-hand side of Fig. 2 shows. These three elements are denominated as the *process context*, and have also been identified as *project management influencers*, according to the ISO 21500 [41] international standard.

As Fig. 2 shows, the context is therefore set out in the tailoring approach. Only the requirements are used to build the solution, and software processes have additionally been considered as the only "live" entities, with the capability of changing and evolving. Additionally, from the VRP viewpoint, the elements surrounding and affecting process enactment are considered as being fixed, and if they change, SPRINTT cannot manage them.

The first solution to make the context an active part of the variant rich process paradigm would be to simply widen it on the left-hand side (Fig. 2). Tailoring requirements are defined from the context, but it is not part of them. They are composed of the changes that these contexts' elements undergo[1]. The context is broken down into elements that categorize all the influencers of software processes, but it is open to include whatever ones its put into practice elicits.

- Projects are the most commonly-changed elements. However, they are not and cannot be considered as isolated entities. Organizations must follow some kind of working lines and their projects may therefore have a lot of similarities.
- Software organizations do not usually change, or at least not in case of traditional software development. However, Global Software Development projects, which involve different organizations, will differ from each other. But since they are all involved in the same project, it may be assumed that they will have some similarities, and that the changes or differences between them may be delimited.

  The last point regarding organizations is simply that of their scope. The organizational unit developing software is sometimes not the software organization itself, but one of its specific departments, or the union of several departments. They are therefore termed as "organizational units".
- Laws do not change frequently, and when they evolve there are still some similarities between them. In the case of global projects, affected by different countries' regulations, similarities between the laws also appear.

Tailoring requirements are defined from these changes, and they remain in process variations. Moreover, the latter are used to manage processes according to their similarities and differences and, what is most important, to take advantage of them. Herein, it would appear to be feasible to manage contexts through their similarities and differences, and considering *contexts* as *change-rich* (which is similar to Variant Rich Processes), by means of the changes between them.

Contexts' elements (project, organizational unit, and laws) are still part of the problem (processes must fit them). But now the solution is built after these changes have taken place, or even better after the more profound change evidence that the management of context offers. Mechanisms are therefore needed to control changes in the context, and to make them evident, the sooner (more abstracted) the better.

This updated perspective signifies that SPRINTT must be embedded in a wider framework that can manage changes in contexts and drive the process variations through the use of the Variant Rich Process paradigm. The *Software Process Institutionalization based on Context Change Evidence* (*SPICCE*) has therefore been created. *SPICCE seeks the promotion of process tailoring on the basis of the changes (and the evidence of these changes) that appear around the process context (project, organizational units and laws), rather than according to these elements as isolated entities*. Fig. 3 presents an overview of SPICCE, and the elements of which it is composed.

The contribution of SPICCE is to propose the use of tailoring mechanisms not by means of considering the actual organization, project or laws (crossed arrows in Fig. 3), but by considering how these elements have changed from the last time that

[1] As regards this research, changes are considered to occur in contexts and variations in processes.

the process was tailored. Changes in the context therefore make the changes that the tailoring would need evident, and thus how to obtain the new tailored process from the previous one.

SPICCE must be supported by a set of techniques, practices and knowledge to manage context and its change and evolution. The challenges of SPICCE are now therefore driven in three different ways.

- First, it needs to model the context.
- Second, it is necessary to determine how evidence of change appears in the context elements. This continues to be the principal challenge, because they are usually implicit.
- Third, evidence of change must be engrained with the rationale management (included in the VRP), with the aim of fully defining the context change and tracing it with process variations.

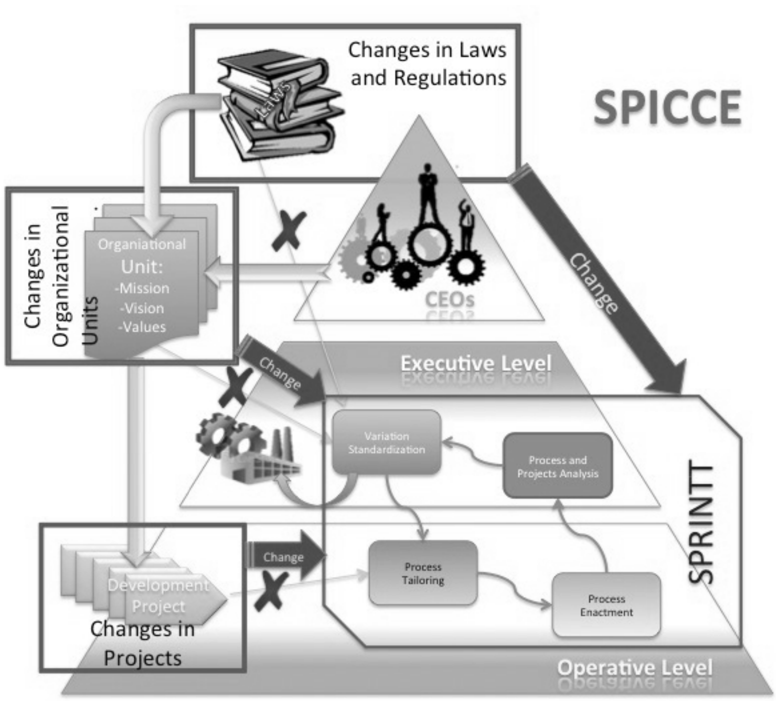


Fig. 3. Software Process Institutionalization based on Context Change Evidence

The following subsections will provide a detailed analysis of how to achieve the three points mentioned above.

3.3 SPICCE Context Modeling Steps

SPICCE needs to model organizational units, laws and projects, and to relate all of them to process variations. Fig. 4 presents a highly abstracted view of the elements of which SPICCE is composed and the standards that will be used to support them. The parts shown in white have already been completed, while those shown in colors are under development or are planned as future work.

The previously developed SPRINTT framework is the core part of SPICCE, and supports process variations, as is described in Section 3.1. The enhancements in SPICCE are divided into three fields. Their results will be engrained with each other in order to configure an ontology [42] modeling the *evidence of context change*.

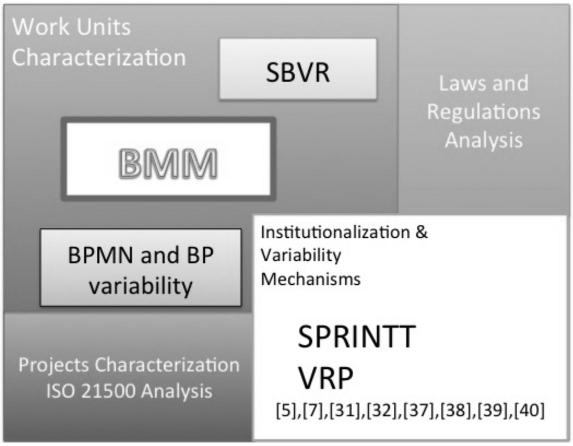


Fig. 4. Work to support SPICCE

- Analysis of Laws and Regulations. Some laws affect software processes. The most common are those concerning Intellectual property, working times, and even some syndical agreements that influence the working day. Changes to laws are, for example, more evident in the case of Global Software Development, but are always in existence.
  Other specific laws are related to desired software functions or characteristics that are under the effects of the project definition.
- Characterization of Projects. Projects are described in various guidelines or regulations, such as the aforementioned Project Management Body of Knowledge (PMBOK) [43], or even the recent ISO standard Guidance on Project Management [41].
- Characterization of the Organizational units. Work units do not have a predetermined size, but are entities with the same objectives. The Object Management Group (OMG) has recently developed several standards, such as the Business Motivation Model (BMM) [9], the Semantics of Business Vocabulary and Business Rules (SBVR) [44], the Business Process Model and Notation (BPMN) [45] or the Organization Structure Metamodel [46], all of which are focused on clearly specifying structure, business rules, strategies, plans, and even business processes (apart from the software processes themselves). All of these standards are commonly used to describe organizations, and therefore affect processes. The BMM has been considered to characterize context, as is described in the following section.

3.4 SPICCE: Eliciting and Representing Changes in Contexts

Changes in process context are not focused on seeking an implementation for them, as occurred with variations. Moreover, context requirements are written in natural language which is informal. More formal representations must therefore be provided to represent context and its changes. In this research, the BMM metamodel and feature models (and FODA –Feature Oriented Domain Analysis [47]) have been considered to represent contexts, break them down and make their changes evident. Feature models present the organization the processes must fit, and the changes that force process tailoring. The following subsections illustrate the approach by considering an application example based on an industrial experience.

Eliciting Changes in the Organizational Units. Organizational units inherit all their characteristics from organizations, and are characterized as them. The Business Motivation Model (BMM) [9] is an OMG proposal whose objective is to structure the development, communicating and managing of business plans in an organized manner. It identifies the elements of business plans, the factors affecting them, and how these are related to each other.

The structure of BMM is depicted in Fig. 5. These sets of elements allow the *means* and *end* to be highlighted, since they describe what the organization is like, or at least, what they wish it to be like; the other important parts are *influencers* and *assessment*. The latter mainly describes how different factors, from inside or outside the organization, affect it. This is particularly important in the case of external influencers which may sometimes be laws. These will be part of a more in-depth analysis of laws and regulations. Assessment depends on each organization, but has no direct effects on the processes or their tailoring.

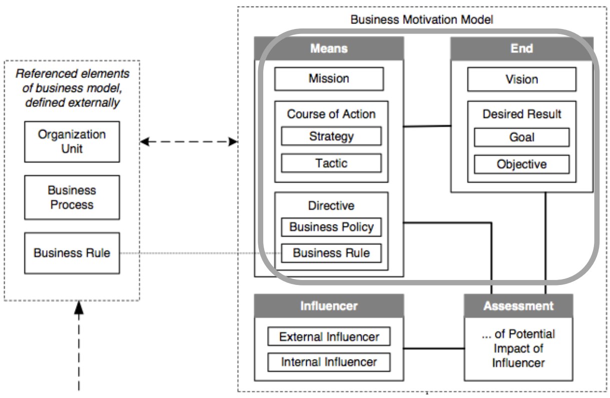


Fig. 5. Structure of BMM

Ends specify how the organizational unit wishes to be at a higher abstraction level, while *Means* include the ways in which to achieve this. Both define the organizational units, and evidence of change may appear in this description. They are decomposed into several children, as Figs. 6 and 7 present. Finally, some relationships have been defined between the ends and means, and their break down structures. As an example, the *mission* implements the *vision*, while *objectives* are related to *practices*. These relationships make it possible to relate abstract elements (in the *ends*) to the concrete practices that support them.

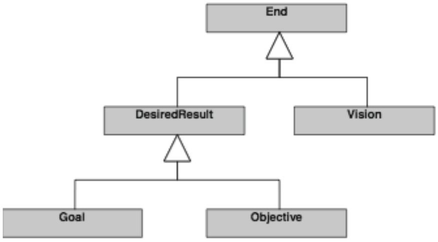


Fig. 6. Hierarchy of *ends* [9]

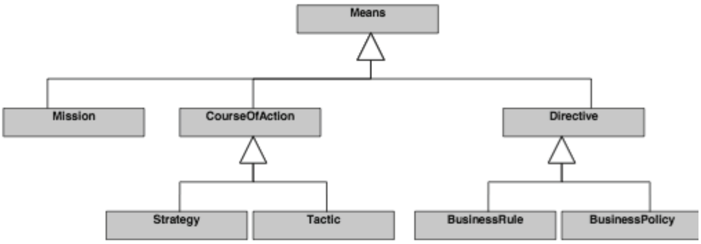


Fig. 7. Hierarchy of *means* [9]

The application study is presented to illustrate how the organization's motivation and its changes affect the process tailoring. This arose after analyzing the results of a previous case study which was conducted to validate VRP in industry.

The application study was executed in the Spanish Alpha software organization[2]. this organization is focused on providing various services, software development, software factories and security, to different clients, banks, public administration, telecommunications, commerce etc. They are experts in several architectures, such as SOA, Client-Server, embedded, business intelligence and so on. Their work is divided into three different departments, *A, B, C*. Each of them has its own version of the same process, as a base process.

They asked us to fuse all their development processes into a single variant-rich process, which would have the ability to be tailored according to different circumstances, and this resulted in the creation of the *Alpha* variant rich process. It also included three crosscutting variations. Each of these tailors the common process according to each of the three departments in the organization:

[2] Real name omitted for reasons of confidentiality.

- Highly complex
- Medium complex
- Medium simple

Some other on-point (specific) variations were also identified.

The variations obtained after executing the case study have been aligned with the organization's motivation, according to the BMM (Table 1). The results shown have been simplified in order to provide a more clear illustration of the proposed approach.

The results in Table 1 were used to align the strategy (in bold type) with the kind of developed projects, and resulted in the kind of tailoring that appears in the *Alpha* variant rich process. The strategy was modeled by using feature models (Fig. 8).

Table 1. Motivation of the organization

	Element	Value
Ends	Vision	To be the leading Spanish software development organization in their market segment.
	Goal	To provide high quality products and services
	Objective	To improve people's qualifications
Means	Mission	To provide software services in Spain
	Strategy	**Efficient and Efficacious management of software development in the** • **Highly complex projects (dep. A)** • **Medium complex projects (dep. B)** • **Medium simple projects (dep. C)**
	Tactic	Contracting highly qualified people who will, by the end of the year, be given a bonus.

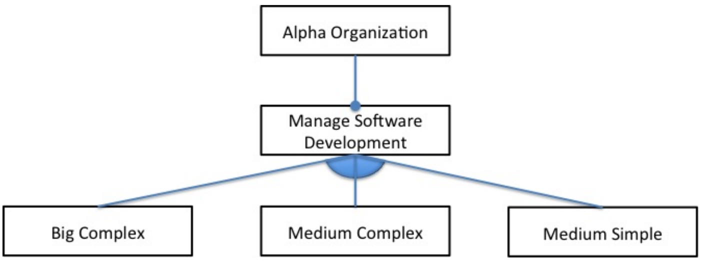


Fig. 8. Feature model of the strategy

It is now easier to see that the organization's motivation is divided into three different sub-points, which are related to the variations needed when tailoring a process.

Tailoring is now simplified to determine how the context has changed, in order to make the correct variations. Fig. 9 shows a graphic depiction of these correspondences. The slashed arrows signify the correspondence between each context change and the variations. They generate the final tailored process.

In order to create a tailored process that fits a "medium complex" project, it is only necessary to choose that change in the context, and the variations supporting it are set automatically. The process is then tailored automatically.

3.5 Engraining Change Evidence and Rationale

Fig. 9 shows a simplified situation in which there is only one change in the context to be taken into account. A real context will imply far more changes, which will occur together. Obtaining the set of variations that best fits each set of changes therefore implies making decisions about them, which is supported by Rationale.

The difference that SPICCE makes to process tailoring is that, instead of analyzing all the tailoring requirements that a context generates, it is only necessary to analyze its context changes. That is, it is only necessary to tailor what has changed.

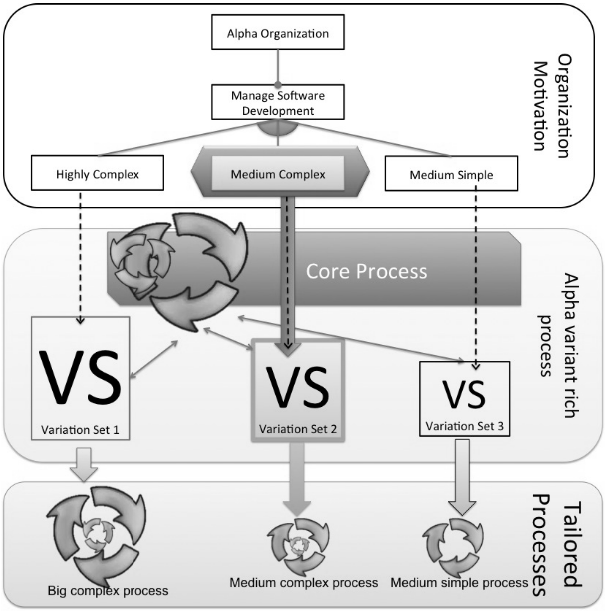


Fig. 9. Tailoring from context change

4 Conclusions and Future Work

Problems must be confronted and solved from the beginning. Existing tailoring mechanisms have attempted to do this, but since the related context causes which can drive variability are omitted, they cannot offer the most suitable solution.

Tailoring signifies that all the characteristics of a process context are included in the process, and tailoring processes according to contexts that are almost the same therefore result in almost the same processes. It would therefore appear to be better to extract the most from the similarities and differences of the contexts, and reflect them in the processes. Although this philosophy implies widening the frontiers of process tailoring to fields that have traditionally been outside this scope, this could provide process tailoring with several advantages. First, tailored processes are strongly linked to their context, and changes in the latter could immediately be traced to the evolution of the processes. Additionally, changes allow process variations to be detected much earlier than when they are needed. These *earlier variations* provide both the variations requirements and the variability support needed.

This article has presented a proposal that is focused on integrating process variations with the actual changes in the context that motivate them. SPICCE provides a set of techniques with which to analyze and decompose each context into projects, organizations – denominated as organizational units-, and laws, and manage them according to their similarities and differences. Bearing in mind that these must later be represented or included in the (tailored) processes, making context changes evident and extending them to the process will deal with the process variations that configure the tailored process.

Implementing SPICCE involves knowing how the context is modeled, how it includes changes, and defining how to turn them into actual variations. Moreover, since this task is difficult owing to the high number of standards that are currently used to describe projects, organizational units and laws, the first step has been focused on analyzing the Business Motivation Model. The study has illustrated the potential usefulness of the proposal with an application example based on a real industrial case. It highlights that context changes that are barely evident sooner or later become process variations, and shows how it is possible to tailor the process from the context by managing these changes.

Our future work lies in complementary directions. First, it would be interesting to execute some extra case studies with which to validate the existing SPRINTT variability mechanisms in different domains, which might additionally provide feedback as regards *changes evidence*. The second direction is focused on completing SPICCE with the analysis of the standards and international regulations, as stated previously. We also plan to homogenize different regulations, such as the BMM and ISO 21500, as they are different viewpoints that represent the same context, and after that, to create an ontology to engrain projects, organizations and laws.

Acknowledgements. This work has been developed under the GEODAS-BC project (Min. de Economía y Competitividad and Fondo Europeo de Desarrollo Regional FEDER, TIN2012-37493-C03-01).

References

1. Zahran, S.: Software Process Improvement: Practical Guidelines for Business Success. Addison-Wesley, Harlow (1998)
2. Rombach, H.D.: Integrated Software Process and Product Lines. In: Li, M., Boehm, B., Osterweil, L.J. (eds.) SPW 2005. LNCS, vol. 3840, pp. 83–90. Springer, Heidelberg (2006)
3. Sutton Jr., S.M.: Aspect-Oriented Software Development and Software Process. In: Li, M., Boehm, B., Osterweil, L.J. (eds.) SPW 2005. LNCS, vol. 3840, pp. 177–191. Springer, Heidelberg (2006)
4. Simidchieva, B.I., Osterweil, L.: Categorizing and Modeling Variation in Families of Systems: a Position Paper. In: Gorton, I., Cuesta, C.E., Ali Babar, M. (eds.) Proc. of the 4th European Conf. on Software Architecture: Companion, pp. 316–323. ACM, Copenhagen (2012)
5. Martínez-Ruiz, T., García, F., Piattini, M.: Supporting Software Process Variability: The Variant Rich Process Modelling Paradigm Sent to JSEP (2012)
6. Brinkkemper, S.: Method Engineering: Engineering of Information Systems Development Method and Tools. Information and Software Technology 38(4), 275–280 (1996)

7. Martínez-Ruiz, T., García, F., Piattini, M., Münch, J.: Modeling Software Process Variability: An Empirical Study. IET Software 5(2), 172–187 (2011)
8. Martínez-Ruiz, T., García, F., Piattini, M.: Process Institutionalization Using Software Process Lines. In: Cordeiro, J., Fillipe, J. (eds.) 11th Int. Conf. on Enterprise Information Systems, ISAS, Milan, pp. 359–362 (2009)
9. OMG: Business Motivation Model. Object Management Group (2012)
10. Sutton, S., Osterweil, L.J.: PDP: Programming a Programmable Design Process. In: Int. Workshop on Software Specification and Desing, pp. 186–190. Schloss Velen (1996)
11. Sutton, S.M., Osterweil, L.J.: Product Families and Process Families. In: 10th International Software Process Workshop (ISPW 2006). IEEE CS (1996)
12. Martínez-Ruiz, T., Münch, J., García, F., Piattini, M.: Requirements and Constructors for Modeling Variability in Software Processes, a Systematic Review. Software Quality Journal 20(1), 229–260 (2012)
13. Araujo, F., Freire, M.A., Câmara dos Santos, W., Kulesza, U.: An Approach to Manage and Customize Variability in Software Processes. In: 24th Brazilian Symposium on Software Engineeering, SBES 2010, pp. 118–127. IEEE Digital Library, Salvador (2010)
14. Hurtado Alegría, J.A., Bastarrica, M.C., Quispe, A., Ochoa, S.F.: An MDE Approach to Software Process Tailoring. In: Proceedings of the 2011 International Conference on Software and Systems Process, pp. 43–52. ACM, Waikiki (2011)
15. Simmonds, J., Bastarrica, M.C.: Modeling Variability in Software Process Lines. Departamento de Ciencias de la Computación. Universidad de Chile (2011)
16. Martins, P.V., Silva, A.R.: ProPAM: Discussion for a New SPI Approach. Software Quality Professional 11(2), 4–17 (2009)
17. Martins, P.V., Silva, A.R.: ProPAM. SPI based on Process and Project Alignment. In: Khosrow-Pour, M. (ed.) Managing Worldwide Operations and Communications with Information Technology. IDEA Group Inc., Vancouver (2007)
18. Killisperger, P., Stumptner, M., Peters, G., Grossmann, G., Stückl, T.: Meta Model Based Architecture for Software Process Instantiation. In: Wang, Q., Garousi, V., Madachy, R., Pfahl, D. (eds.) ICSP 2009. LNCS, vol. 5543, pp. 63–74. Springer, Heidelberg (2009)
19. Silva Barreto, A., Murta, L., Rocha, A.R.: Software Process Definition: a Reuse-Based Approach. In: XXXIV Conf. Latinoam. de Informática, pp. 409–418 (2008)
20. Silva Barreto, A., Murta, L., Rocha, A.R.: Software Process Definition: a Reuse-Based Approach. Journal of Universal Computer Science 17(13), 1765–1799 (2011)
21. Lu, R., Sadiq, S.K.: Managing Process Variants as an Information Resource. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 426–431. Springer, Heidelberg (2006)
22. Lu, R., Sadiq, S.: A Reference Architecture for Managing Business Process Variants. In: 9th ICEIS 2007 (2007)
23. Liu, J.Y.-C., Chen, V.J., Chan, C.-L., Lie, T.: The impact of software process standardization on software flexilblity and project management performance: Control theory perspective. Information and Software Technology 50, 889–896 (2008)
24. Mezini, M., Ostermann, K.: Variability Management with Feature-Oriented Programming and Aspects. ACM SIGSOFT Software Engineering Notes 29(6), 127–136 (2004)
25. Charfi, A., Schmeling, B., Mezini, M.: Transactional BPEL Processes with AO4BPEL Aspects. In: 5th European Conf. on Web Services, ECOWS 2007, pp. 149–158. IEEE CS, Halle, Saale (2007)
26. Charfi, A., Mezini, M.: AO4BPEL: An Aspect-oriented Extension to BPEL. Journal of World Wide Web 10(3), 309–344 (2007)
27. Gröner, G., Boskovic, M., Parreiras, F.S., Gasevic, D.: Modeling and validation of business process families, Information Systems. Information Systems (2012)

28. Ploesser, K., Peleg, M., Soffer, P., Rosemann, M., Recker, J.: Learning from Context to Improve Business Processes. BP Trends (2009)
29. Rosemann, M., Recker, J., Flender, C.: Contextualization of Business Processes. Int. J. Business Process Integration and Management 1, 111–127 (2007)
30. Hartmann, H., Trew, T.: Using Feature Diagrams with Context Variability to Model Multiple Product Lines for Software Supply Chains. In: 12th International Software Product Line Conference, pp. 12–21 (2008)
31. Martínez-Ruiz, T., García, F., Piattini, M., de Lucas-Consuegra, F.: Process Variability Management in Global Software Development. A case study. In: ICSSP 2013. Springer, San Francisco (accepted, 2013)
32. Martínez-Ruiz, T., García, F., Piattini, M.: Framework to the Standardization and Institutionalization of Software Processes (2008)
33. Clements, P., Northrop, L.: Software Product Lines. Practices and Patterns. Addison-Wesley, Boston (2002)
34. Filman, R.E., Elrad, T., Clarke, S., Aksit, M.: Aspect-Oriented Software Development. Addison-Wesley, Boston (2004)
35. Dutoit, A.H., McCall, R., Mistrík, I., Paech, B.: Rationale Management in Software Engineering: Concepts and Techniques. In: Dutoit, A.H., McCall, R., Mistrík, I., Paech, B. (eds.) Rationale Management in Software Engineering, pp. 1–45. Springer, Heidelberg (2006)
36. OMG: Software Process Engineering Metamodel Specification. Object Management Group (2008)
37. Martínez-Ruiz, T., García, F., Piattini, M.: Managing Process Diversity by Applying Rationale Management in Variant Rich Processes. In: Caivano, D., Oivo, M., Baldassarre, M.T., Visaggio, G. (eds.) PROFES 2011. LNCS, vol. 6759, pp. 128–142. Springer, Heidelberg (2011)
38. Martínez-Ruiz, T., García, F., Piattini, M., Münch, J.: Applying AOSE Concepts for Modeling Crosscutting Variability in Variant-Rich Processes. In: Biffl, S., Koivuluoma, M., Abrahamsson, P., Oivo, M. (eds.) 37th Conf. on Software Engineering and Advanced Applications, SEAA, pp. 334–338. IEEE, Oulu (2011)
39. Martínez-Ruiz, T., García, F., Piattini, M.: Enhanced Variability Mechanisms to Manage Software Process Lines. In: 16th European Systems & Software Process Improvement and Innovation, pp. 12.13-12.23. Publizon, Alcalá de Henares, Madrid (2009)
40. Martínez-Ruiz, T., García, F., Piattini, M.: Towards a SPEM v2.0 Extension to Define Process Lines Variability Mechanisms. In: Lee, R. (ed.) Software Engineering Research, Management and Applications. SCI, vol. 150, pp. 115–130. Springer, Heidelberg (2008)
41. ISO/IEC: Guidance on Project Management. vol. ISO 21500:2012(E). ISO IEC, Switzerland (2012)
42. Calero, C., Ruiz, F., Piattini, M. (eds.): Ontologies for Software Engineering and Software Technology. Springer (2006)
43. Guide, P.A.: to Project Management Body of Knowledge. Proyect Management Institute (2000)
44. OMG: Semantics of Business Vocabulary and Business Rules Object Management Group (2008)
45. OMG: Business Process Modeling Notation Specification. OMG (2006)
46. OMG: Organization Structure Metamodel Specification. Object Management Group (2009)
47. Kang, K.C., Cohen, S.G., Hess, J.A., Novak, W.E., Peterson, A.S.: Feature-oriented domain analysis (FODA) feasibility study. SEI, Carnegie Mellon University (1990)

Towards a Formal Approach for Prototyping and Verifying Self-Adaptive Systems

Juan F. Inglés-Romero[1] and Cristina Vicente-Chicote[2]

[1] Dpto. Tecnologías de la Información y Comunicaciones,
Universidad Politécnica de Cartagena, Edificio Antigones, 30202 Cartagena, Spain
juanfran.ingles@upct.es
[2] Quercus Software Engineering Group (QSEG),
Universidad de Extremadura, Avda. de la Universidad S/N, 10003 Cáceres, Spain
cristinav@unex.es

Abstract. Software adaptation is becoming increasingly important as more and more applications need to dynamically adapt their structure and behavior to cope with changing contexts, available resources and user requirements. Maude is a high-performance reflective language and system, supporting both equational and rewriting logic specification and programming for a wide range of applications. In this paper we describe our experience in using Maude for prototyping and verifying self-adaptive systems. In order to illustrate the benefits of adopting a formal approach based on Maude to develop self-adaptive systems we present a case study in the robotics domain.

Keywords: Self-Adaptive Systems, Prototyping, Maude, VML.

1 Introduction

Nowadays, significant research efforts are focused on advancing the development of (self-) adaptive systems. In spite of that, some major issues remain still open in this field [1][2]. One of the main challenges is how to formally specify, design, verify, and implement applications that need to adapt themselves at runtime to cope with changing contexts, available resources and user requirements.

Adaptation in itself is nothing new, but it has been generally implemented in an ad-hoc way, that is, developers try to predict future execution conditions and embed the adaptation decisions needed to deal with them in their application code. This usually leads to increased complexity (business logic polluted with adaptation concerns) and poor reuse of adaptation mechanisms among applications [1]. The use of formal methods can help alleviating the limitations of current approaches to self-adaptive system development. In particular, they can provide developers with rigorous tools for testing and assuring the correctness of the adaptive behavior of their systems. This is a remarkable open issue, since only a few research efforts seem to be focused on the formal analysis and verification of self-adaptive systems.

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 432–446, 2013.

Maude [3] is a high-performance reflective language and system supporting both equational and rewriting logic specification and programming for a wide range of applications. Maude and its supporting tools can be used in three, mutually reinforcing ways: as a declarative programming language, as an executable formal specification language, and as a formal verification framework. A Maude program can be seen as an executable mathematical model of a system. Thus, using Maude for prototyping self-adaptive systems enables their simulation, formal analysis (e.g., reachability/likelihood of certain system configurations), and verification (e.g., testing that the system reaches a consistent configuration for all given contexts). Furthermore, Maude can help designers to assure, among other properties, the consistency and correctness of self-adaptive system specifications.

This paper presents our experience in using Maude for prototyping and verifying component-based self-adaptive systems. The main contribution derives from the implementation of a robotic case study, in which we model the variability of the system and manually translate this logic into Maude. This experience has allowed us to witness some of the benefits of adopting a formal approach when developing self-adaptive systems. This research continues our previous works on self-adaptive system design [4] and implementation [5] using the DiVA framework [6], and on modeling self-adaptive system variability using the *Variability Modeling Language* (VML) [7].

2 Robotic Example

In this section, we introduce the robotic example designed to illustrate the versatility of Maude for prototyping self-adaptive systems. It is worth noting that, although the example presented here belongs to the robotics domain, in which the need for self-adaptation (to changing contexts and available resources) is quite obvious, we believe that our proposal is general enough to be adopted in other application domains in which modeling, simulating and formally verifying self-adaptation is an issue.

2.1 Adaptation Scenario

The proposed case study takes place in a room, where a small robot moves around randomly avoiding obstacles. In order to improve this basic functionality in terms of safety, power consumption and performance, the robot follows an adaptation strategy that decides on the following variation points: (1) the signaling type; (2) the signaling intensity; and (3) the robot velocity. There are two possible variants for the signaling type (light or acoustic), while the signaling intensity and the robot velocity may take any integer value in the range 0-100. The adaptation strategy decides the best possible configuration (selection of variants for each variation point) according to the current context. The context variables considered in the case study are the ambient light, the ambient noise and the robot battery level, all of them integers ranging from 0 to 100.

The goodness of each configuration is calculated based on the impact of each variant on the three properties being considered, that is: safety, power consumption and performance. The following considerations are made concerning *safety* (making others aware of the presence of the robot in the surroundings): (1) light signaling is more convenient than acoustic signaling when the ambient light is low; and (2) the higher the ambient noise (might indicate a crowded environment), the higher must be the signaling intensity and the lower the robot velocity. Regarding *power consumption*, the greater the signaling intensity and the robot velocity the greater the power consumption. Thus, if the battery level is low, both the velocity and the signaling intensity need to be limited. Finally, concerning *performance*, the higher the velocity the shorter the time it takes to the robot to reach its goal position. Obviously, maximizing safety and performance, while simultaneously minimizing power consumption, imposes conflicting requirements. Thus, the adaptation strategy will need to find the right balance among these requirements to achieve the best possible configuration for a given context, even if some (or none) of them are optimized individually.

2.2 Component-Based Software Architecture

The component-based software architecture developed for the case study is sketched in Figure 1. As other self-adaptive systems [2], the proposed design includes: (1) a reconfigurable part, comprising the optional and/or parameterized components; (2) a set of monitoring components; and (3) an adaptation control unit.

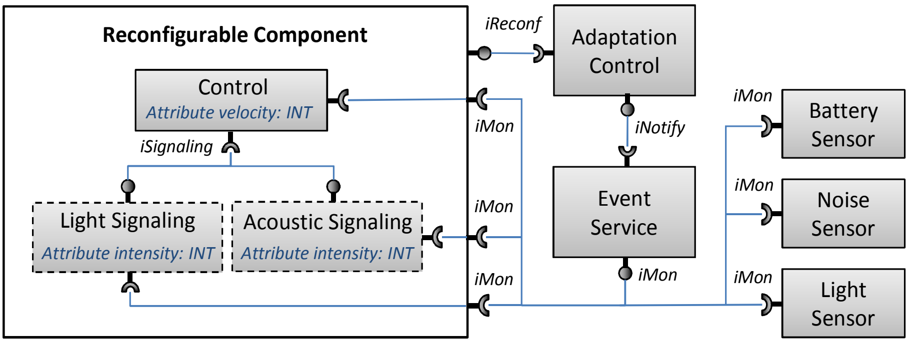


Fig. 1. Component-based software architecture for the example

The *Reconfigurable Component* gathers the elements of the system that are susceptible to change at runtime. Among them, the *Control Component* implements the core robot functionality, that is, the motion control and the obstacle avoidance. This component includes a parameter called *velocity* that regulates the robot motion speed, and is responsible for activating or deactivating the robot signaling through the *iSignaling* interface. The *Reconfigurable Component* also contains two optional components, each one implementing one of the alternative ways for signaling the robot position: *Light Signaling* and *Acoustic Signaling*. Both these components contain an *intensity* parameter that regulates the frequency of the light and the acoustic signals, respectively. The three variation points available at the *Reconfigurable Component* (i.e., selecting one of the two alternative signaling components and setting the velocity and the intensity parameters) will need to be fixed at runtime by the adaptation strategy, implemented by the *Adaptation Control*, as detailed later.

The monitoring part of the architecture provides the context-aware support for the adaptation. It comprises (1) a set of sensors (*Noise Sensor*, *Light Sensor* and *Battery Sensor*) and monitors (*Control*, *Light Signaling* and *Acoustic Signaling*) for acquiring information both from the environment (external context) and from the system itself (internal context); and (2) the *Event Service* component that receives the context information from the former components via the *iMon* interface, and notifies the changes in the context to the *Adaptation Control* component through the *iNotify* interface.

Finally, the *Adaptation Control* component implements the adaptation strategy which, on the basis of the context changes notified by the *Event Service* component, decides which is the best possible configuration (variant selection) for the *Reconfigurable Component* and applies the required changes via the *iReconf* interface.

We have selected the E-puck robot [8] as our target platform. E-pucks are low-cost mobile robots with a large range of sensors and actuators that make them appropriate for testing self-adaptive applications. However, this robot presents limitations that prevent executing Maude on it. For this reason, we use a distributed architecture where some components are deployed in an external PC, which communicates with the robot via Bluetooth (e.g., this is the case of the Adaptation Control, which relies on Maude for executing the adaptation strategy). As our intention is to use Maude for prototyping the system (as a support for its design, simulation, and verification) and not for its final implementation, this seems to be a valid approach.

3 Modeling Variability with VML

This section introduces the *Variability Modeling Language (VML).* VML provides a mechanism to express how a system should adapt at run-time to improve its performance under changing conditions. The current version of VML has been developed using a Model-Driven Engineering (MDE) approach. We have created a textual editor for VML using the Xtext framework [9], including some advanced features such as syntax checking and coloring facilities, and a completion assistant. As the focus of this paper is not VML, we will only present the essentials needed for modeling the case study. Then, the resulting VML model will be used as the starting point for obtaining the Maude specification included in Section 4.

In a VML model, we first need to define the ***variation points***. Aligned with *Dynamic Software Product Lines (DSPL)* [10], VML variation points represent points in the software where different variants might be chosen to derive the system configuration at run-time. Therefore, variation points determine the decision space of VML, i.e., the answer to *what* can change. As shown in Listing 1, variation points (`varpoint`), as all the other VML variables, belong to a certain data type. VML includes three basic data types: enumerators, ranges of numbers, and booleans. For instance, the *velocity* variation point is an enumerator that set the robot speed using three possible variants: SLOW, NORMAL or FAST. After defining the variation points, we need to specify the ***context variables*** (`context`), which identify the situations in which variation points need to be adapted. Listing 1 shows three context variables: (1) the ambient lighting, (2) the ambient noise, and (3) the battery level. Note that contexts and variation points have been modeled to abstract the original parameters presented in Section 2, most of them defined as integers ranging from 0 to 100. This abstraction reduces the complexity considering only the relevant values for each variable.

At this point, we need to define *how* variation points are set according to the contexts. This is achieved through *properties* (`property`) and *ECA (event-condition-action) rules* (`rule`). ***Properties*** specify the features of the system that need to be optimized, i.e., minimized or maximized. Properties are defined using two functions: *priorities* and *definitions.* Definitions characterize the property in terms of variation points (i.e., definitions are the objective functions to be optimized). For instance, in Listing 1, we define the *performance* property as a linear function of the velocity variation point (the faster the robot accomplishes its task, the better its performance). It is worth noting that property definitions can be characterized using the technical specifications of the hardware (e.g., to know how much power each component consumes), simulation or empirical data from experiments. On the other hand, *priorities* describe the importance of each property in terms of one or more context variables (i.e., priorities weight the objective functions). For instance, *power consumption* becomes more relevant whether the battery level decreases below 20%. Otherwise, *power consumption* is not considered an issue and, as a consequence, its impact on the adaptation process is very small. Opposite to definitions, priorities are characterized in a more subjective way, depending on the designer experience. Regarding the ***ECA rules,*** they define direct relationships between context variables and variation points. As shown in Listing 1, the left-hand side of a rule expresses a trigger condition (depending on one or more context variables) and its right-hand side sets the variation point. For example, the decision of which signaling (acoustic or light) to select in each situation is modeled using rules. Basically, when there is ambient light (lightning = true) the first rule selects the acoustic signaling. Otherwise, the second rule selects the light signaling.

Regarding the execution semantics, VML models specify a constrained optimization problem, that is, they use syntactic sugar for describing the global weight function that optimizes the variation points and, as a result, allows improving the overall system quality. This global function is obtained by aggregating the property definitions (terms to be optimized), weighted by their corresponding priorities. Besides, the ECA rules state constraints that need to be satisfied. At this point, it is worth noting that VML variation points and contexts are high-level variables that somehow abstract architectural elements (e.g., components or component parameters). For instance, the *velocity* is linked to the parameter velocity in *Control* component, and the battery level is obtained from the *Battery Sensor* component (see Figure 1). This abstraction allows VML to be independent of the underlying architecture, what, among other benefits, enables the reuse of the models in different platforms and scenarios. In the next section we start from the VML model, shown in Listing 1, to obtain an equivalent Maude specification.

```
type impact : enum {VERY_HIGH(5), HIGH(4), MEDIUM(3),
                    LOW(1), VERY_LOW(2), NON_EFFECT(0)};
context lighting  : boolean;
context noise     : enum {LOW, MEDIUM, FULL};
context battery   : enum {NORMAL, NOISY, VERY_NOISY};

rule rule1 : lighting = true           => signalType = ACOUSTIC;
rule rule2 : lighting = false => signalType = LIGHT;
rule rule2 : battery <> FULL           => velocity <> FAST;

property safety : impact maximized {
    priorities:
        case noise <> NORMAL : impact.MEDIUM
        default: impact.NON_EFFECT;
    definitions:
        f(signalIntensity) = signalIntensity/20;
        f(velocity) = velocity/20 - 1; }

property powerConsumption : impact minimized {
    priorities:
      case battery = LOW : impact.VERY_HIGH
      default: impact.LOW;
    definitions:
        f(signalIntensity) = signalIntensity/40 + 1.5;
        f(velocity) =  velocity/40 + 1.5; }

property performance : impact maximized {
    priorities:
      case noise <> NOISY and battery = LOW:  impact.VERY_LOW
      default: impact.HIGH;
    definitions:
      f(velocity) = velocity/20;
}
varpoint signalType          : enum {ACOUSTIC, LIGHT};
varpoint signalIntensity     : enum {LOW(20), MEDIUM(60), HIGH(100)};
varpoint velocity            : enum {SLOW(20), NORMAL(60), FAST(100)};
```

Listing 1. Variability model described using VML for the robotic example

4 Prototyping Self-Adaptation with Maude

In order to learn about the convenience of using formal tools like Maude in the development of self-adaptive systems, this section describes the Maude specification for the adaptation logic modeled in Section 3. Note that, for the lack of space, we do not provide the complete Maude specification, but only the essential concepts for modeling the self-adaptation logic.

4.1 Overall Proposed Approach

We have implemented our example with Core Maude using an object-based programming approach. This allows us to model self-adaptive systems as configurations (collections) of objects and messages that represent (a snapshot of) a possible system state. Each object has an identifier, a class, and a set of attributes. For instance, the expression `< oid : cid | attr1, attr2 >` represents an object with identifier `oid`, belonging to the class `cid`, and with two attributes `attr1` and `attr2`. On the other hand, messages include an identifier and a list of arguments. For example, the expression `mid (arg0, arg1)` represents a message with identifier `mid` and arguments `arg0` and `arg1`. The idea behind using a set of objects and messages to represent the system state is that we can specify the adaptation behavior as a set of rewrite rules that consume and produce objects and messages, i.e., that evolve the system state.

As in most self-adaptive systems [2], the adaptation loop comprises three processes, namely: (1) gathering and assessing the current context, (2) reasoning on the best adaptation possible, and (3) performing the system reconfiguration. In our case, Maude carries out all these processes through the *Adaptation Control* component, which simply allows Maude to interact with the component architecture. The interface between Maude and the *Adaptation Control* is summarized in Table 1. Figure 2 outlines the steps of the algorithm that implements this adaptation loop. Each of these steps is further detailed in the following subsections.

Table 1. Interface between Maude and the *Adaptation Control* component

Command	**Description**
Start	Starts the adaptation loop
synchArch <component : String> <parameter : String> <value : String>	Synchronizes the Maude architecture representation with the actual architecture implementation. Example: *synchArch "LightSignaling" "state" "running"* → Maude is notified of the actual state of the *LightSignaling* comp.
init (<battery : INT>, <noise : INT>, <light : INT>)	Maude is notified of the initial low-level context variables Example: *init (100, 55, 20)*
battery <value : INT>	Updates the battery (0-100). Example: *battery 23*
noise <value : INT>	Updates the ambient noise (0-100) Example: *noise 67*
light <value : INT>	Updates the ambient light (0-100) Example: *light 10*
notify <component : String> <parameter : String> <value : String>	Maude is notified of a change in a component. Example: *notify "control" "velocity" "23"* → The control component notifies that the velocity has changed to 23
command <component : String> <parameter : String> <value : String>	Maude sends a reconfiguration command. Example: *command "control" "velocity" "11"* → The velocity of control component must be changed to 11

4.2 Initialization

At run-time, Maude needs to keep the dynamic state of the adaptation process through an internal representation (or model) of: (1) the *current context*, to know when a new situation (e.g., a new battery level) produces significant changes to require an adaptation, and (2) the *current component architecture*, to trace what modifications are needed to obtain an adapted configuration of the system from the current one. Consequently, prior to starting the adaptation loop, an initialization function needs to set up the context and the architecture representations. This function is labeled in Figure 2 as "*Init context and synchronize representation*".

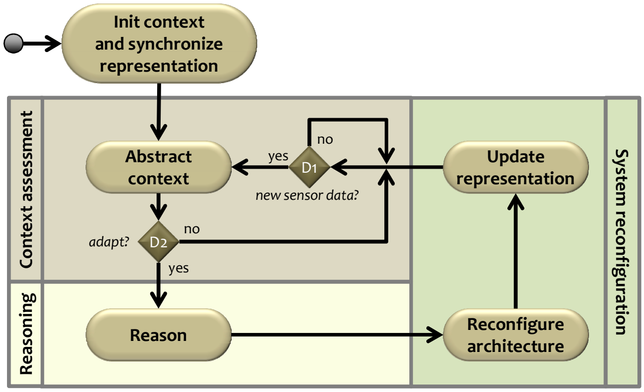


Fig. 2. Outline of the adaptation loop

The Context Representation. It should be both detailed enough to gather all the contextual information relevant for the adaptation, and abstract enough to enable the system to efficiently reason on it. This representation considers the same description as the one given for the context variables in the VML model (see Listing 1). The function that computes the values from the components to obtain a more abstract representation (e.g., deciding when the integer provided from the *Battery Sensor* is LOW, MEDIUM or FULL) will be later detailed in section 4.3. A possible configuration of the context model in Maude could be as follows:

```
< ctx : Context | batt  : FULL, noise : NORMAL, light : false >
```

The Architecture Representation. Similarly to the context model, maintaining an explicit reflection model that abstracts the actual running system is essential to efficiently decide on it and execute the required reconfigurations. This model needs to be synchronized with the actual component-based system architecture in order to provide the adaptation logic with up-to-date information. With regard to adaptation, the only relevant information contained in the system architecture for the example (see Figure 1) is the list of components gathered in the *Reconfigurable Component* (neither the component interfaces nor the connectors are modeled). Each component in this list is modeled in Maude with an object containing, at least, two attributes: *name* (String) and *state* $\in$ {*RUNNING*, *STOPPED*}. An additional attribute will be added for each parameter defined in each component. A possible configuration of the architecture model could be as follows (please, note that the state of the AcousticSignaling component is STOPPED, meaning that it is not present in the actual system architecture):

```
<c : Control | name : "control", state : RUNNING, velocity : 5  >
<l : LightSignaling | name : "lsig", state : RUNNING, intensity : 50 >
<a : AcousticSignaling | name : "asig", state :STOPPED, intensity:50>
```

The initialization process is addressed through two rewrite rules. On the one hand, the `arch-synchronization` rewrite rule is triggered each time the Adaptation Control component sends to Maude a `synchArch` message (see Table 1), which updates the state and attributes of the corresponding component in the architecture representation. This occurs when the components belonging to the *Reconfigurable Component* initially notify their state via *iMon* interface. On the other hand, the `init-context` rewrite rule updates the context representation after receiving the measurements of all sensors through an `init` message (see Table 1).

4.3 Context Assessment

The main functions of the *Context Assessment* process are: (1) to receive the context information from the sensors (see the `battery`, `noise` and `light` commands in Table 1), (2) to translate this raw data into consistent values for the more abstract context representation, and (3) to launch the reasoning process in case the changes in the context variables are significant enough. These three functions are represented in Figure 2 in the decision node *D1*, the operation "*Abstract context*" and the decision node *D2*, respectively.

In order to address the context assessment, we have implemented three rewrite rules (one for each context variable). Thus, a new message from a sensor will trigger its corresponding rule (1) to apply an abstraction based on fixed thresholds (e.g., we consider the *battery* to be *FULL* when we receive a value greater than 80), and (2) to determine whether an adaptation is required according to how much the context has changed. In the current implementation, the adaptation process starts only if a variable in the context representation changes. In this case, a `reasoner` message is created to trigger the rules performing the reasoning process, as detailed next.

4.4 Reasoning

The *Reasoning* function (see Figure 2) implements the core self-adaptation logic as it computes the best configuration possible for a given context, that is, it selects the set of variants that jointly optimize the overall system performance. The Maude specification does not only rely on the description of the properties and rules described in the VML model, but also on their semantics. Therefore, as we mentioned in Section 3, Maude will need to solve the underlying constrained optimization problem described in the VML model.

The Variability Representation. It translates the variation points included in the VML model into Maude. As shown below, each variant is modeled in Maude with an object containing the following attributes: *name*, *dimension* (ID of the variation point the variant belongs to), *safety*, *consumption* and *performance* (impact of the variant in each property resulted from evaluating the definition function), *score* and *state*. Note that, as in the VML model, we abstract the architecture details. For instance, whereas *signaling intensity* is a component attribute in the range 0-100, both in VML and in Maude, we only consider three variants (*LOW, MEDIUM, HIGH*). This abstraction, together with the one provided by the context and architecture representations, significantly simplifies the reasoning process.

```
< v : Variant | name : "slow", dimension : "velocity", safety : 3,
  consumption : 2, efficiency : 1, score : 0, state : AVAILABLE >
```

The reasoning approach followed in this research is based on the method described in [11], which combines the use of adaptation rules and the optimization of property-based adaptation goals. Our adaptation rules have been implemented as two Maude rewrite rules, `non-available` and `required`. Both these rules are executed once for each variant object, updating its *state* attribute according to the current context. The `non-available` rule sets the *state* of those variants that are inconsistent with the current context (i.e., cannot be selected during the subsequent optimization process) as NON-AVAILABLE. For example, if the *battery* is not *FULL*, then the variant *FAST* is marked as NON-AVAILABLE (see *rule3* in the VML model). The `required` rule sets the *state* of those variants that, according to the current context, need to be compulsorily selected as REQUIRED. For example, if *light* is *true*, then the variant *ACOUSTIC* is marked as REQUIRED (see *rule1* in the VML model).

In order to cope with the optimization of property-based adaptation goals, we have implemented two additional rewrite rules: `calculate-scores` and `search-solution`. The first of these rules is triggered once for each variant and calculates the attribute *score* of those marked as AVAILABLE. The calculation of the *score* is based on: (1) the impact of each variant on the three system properties (evaluation of the function for the property *definition*); and (2) the importance of each property in the current context (the evaluation of the function for the property *priority*). Finally, the `search-solution` rule finds the best possible system configuration for the current context, i.e., the combination of variants that, together, obtain the highest score.

4.5 System Reconfiguration

The main functions of the *System Reconfiguration* process are: (1) to create a reconfiguration plan (sequence of reconfiguration commands) that adapts the architecture representation according to the decisions made by the *Reasoning* function, and (2) to synchronize the architecture representation with the run-time system architecture. To implement these functions, labeled in Figure 2 as "*Reconfigure architecture*" and "*Update representation*", we have implemented two Maude rewrite rules: `reconfigure` and `notification-when-pending`.

The `reconfigure` rule produces a set of reconfiguration commands (see *command* in Table 1) for those components that need to be modified (i.e., those for which the state or other attribute has changed). This is achieved by making the difference between the current architecture representation and the one that has just been derived from the selected variants. Moreover, the `notification-when-pending` rule is executed whenever a real component (belonging to the *Reconfigurable Component*) notifies that it has changed in response to a reconfiguration command (see *notify* in Table 1). These notifications cause the architecture representation to be updated to reflect the current situation. It is worth noting that Maude registers all the reconfiguration commands sent and not acknowledged yet. Context messages are discarded as long as there are pending notifications. This prevents the execution of new adaptation loops while the architecture representation is not completely synchronized.

5 Formal Verification of the Adaptation Strategy

Maude can be used not only for simulating self-adaptation strategies, but also for formally checking that the specification satisfies some important properties under certain conditions, or for obtaining useful counterexamples showing that some property is violated. As this kind of model-checking analysis can be a powerful tool for self-adaptive system designers, this section illustrates how Maude allows checking invariants using the *search* command.

As previously mentioned, the *Adaptation Control* component (see Figure 1) relies on Maude for executing the adaptation strategy. Thus, sensor components interact (indirectly) with Maude to provide context information (e.g., the battery level or the ambient noise). Although this scheme is appropriate for simulating or running the system, it is not for exhaustively checking whether the model satisfies a given condition. Therefore, we need to automatically generate context configurations to enable the Maude *search* command to explore the adaptation space. For instance, in the proposed example, the robot should not run simultaneously the light signaling and the acoustic signaling components. We can verify that this property holds in all situations using the *search* command as shown below.

```
search in SIMULATOR :
    start
=>+
    contextModel(<ctxObj : Context| light:LIGHT,noise:NOISE,batt:BATT>)
    archModel(
      < ctrlObj : Control | ATTS-CFG >
      < lSigObj : LightSignaling | state : LSIG-ST, ATTS-CFG' >
      < aSigObj : AcousticSignaling | state : ASIG-ST, ATTS-CFG" > )
    CONF:Configuration
such that (LSIG-ST == RUNNING) and (ASIG-ST == RUNNING) .
```

Given an initial configuration (declared before the arrow), this command explores the adaptation space looking for the pattern that has to be reached (declared after the arrow). Firstly, the starting configuration is the message *start* that activates the context generator and the adaptation loop (see Figure 2). Then, Maude searches for the context (i.e., the value for the variables LIGHT, NOISE and BATT) and the architecture that are set when the *state* of both the *LightSignaling* and the *AcousticSignalig* is RUNNING. If the search command returns no solution it means that on the initial state, and on all states reachable from it, the predicate is an invariant. Similarly, the command could also be used to analyze the reachability of a valid configuration, for example, to find out under which contexts the adaptation selects a velocity greater than 80.

6 Related Work

Significant research efforts are being invested to try overcoming the limitations of current ad-hoc approaches to (self-) adaptive system development. These efforts have given rise to new adaptation-enabling frameworks and middlewares, and new languages supporting adaptation primitives [2]. However, most current approaches do not offer either a formal specification of the adaptation processes, nor a formal reasoning support for testing, assessing and verifying the adaptation logic. This issue has been highlighted as a major challenge in several works [1][2].

Some of the existing frameworks provide some kind of support for self-adaptive system simulation and verification. For instance, as part of the DiaSuite Project [12], the DiaSpec Domain Specific Language (DSL) enables the specification of Sense/Compute/Control (SCC) applications, where the interaction with the physical environment is essential. Basically, DiaSpec allows designing applications independently of the platform, describing entities (e.g., components or devices) and controllers, which execute actions on entities when a context situation is reached. DiaSpec models can be simulated to test the SCC applications before their deployment. These models can be simulated using DiaSim without requiring any coding effort. DiaSim provides a graphical editor to define simulation scenarios and a 2D-renderer to monitor simulated applications. The main benefit of this approach is the graphical representation of the environment, where the user can easily see the interactions between entities. However, this approach is quite limited as it does not enable to extensively explore the possible states of the system to verify invariants, find out erroneous reconfigurations, or prove the correctness of the rules.

MUSIC [13] is a framework that supports self-adaptation in mobile applications based on component-based architectures. MUSIC proposes a methodology where the contextual information is modeled extending an ontology, and the architecture reconfigurations are expressed via goal policies, through utility functions in the architectural elements. MUSIC provides a tool for static validation aimed to detect errors and omissions in the specification (e.g., for type checking). On the other hand, MUSIC also offers a simulation tool that enables developers to observe and analyze the effects of context changes and adaptations. In contrast to our approach, where we model separately the application logic and the variability involved in the adaptation process, this approach is too coupled to the underlying architecture.

In the field of Dynamic Software Product Lines (DSPL) [10], MOSKitt4SPL [14] enables designers to model dynamic variability by means of (1) feature models, describing the possible configurations to which the system can evolve, and (2) resolution models, defining the reconfigurations in terms of feature activation/deactivation associated with a context condition. These specifications are automatically transformed into state machines representing the Adaptation Space, where the states are the possible system configurations and the transitions the migration paths. This approach enables to analyze the Adaptation Space and automatically refine the model specifications to ensure the following behavioral issues: (1) determinism, (2) reversibility, (3) absence of redundancy, and (4) nonexistence of cycles. The main advantage of this approach is the possibility of using the well-established Finite State Machine (FSM) theory to analyze the system specification. However, it does not rely on a formal framework like Maude to implement specific verification algorithms or the simulator.

Also in this line, the DiVA [6] Project provides a tool-supported methodology with an integrated framework for managing dynamic variability. DiVA proposes an early validation at design-time to discover faults in the adaptation specification using simulation, i.e., the user manually selects relevant contexts and then analyzes the results from the adaptation logic. Although DiVA has multiple solvers available both for simulating and running the system, this approach seems to be insufficient to check invariants in large models with many context variables or to prove the consistency and correctness of the specification.

7 Lessons Learned and Future Research Plans

To this point, we have presented our experience in using Maude for prototyping and verifying component-based self-adaptive systems. Next, we draw some conclusions and lessons learned from this experience, and outline our future research plans.

Roles of Formal Tools in the Development of Self-Adaptive Systems

At design-time, formal tools, like Maude, can support self-adaptive system modeling, at least, in two ways. On the one hand, they can be used to check the ***semantic correctness*** of the variability models. For instance, considering the VML language presented in section 3, Maude can help checking data types (e.g., we cannot assign a Boolean value to an enumerated variable), detecting unused elements (e.g., a variation point that is never used), avoiding recursive assignments (e.g., x=x+1), or preventing contradictory statements (e.g., two incompatible rules). On the other hand, formal tools allow ***verification*** and ***simulation*** of the adaptation logic. For example, Maude provides the `search` command, which explores the reachable state space looking for a given configuration. This command is a simple, yet powerful method for checking invariants, as introduced in section 5. Furthermore, Maude enables the simulation of the system specification starting from any given state. This can be very useful for adjusting some parameters of the adaptation logic.

At run-time, we can apply formal tools to ***validate the system reconfigurations*** before they are actually performed, as it is not reasonable to make the system migrate to an invalid configuration. For example, DiVA [6] applies an online validation with Kermeta [15] at run-time to check that all the invariants hold. Apart from that, formal tools can also be considered as a means to ***operate with models*** in the adaptation process. The main benefit of this approach is that the implementation of the operations can be verified. For instance, as mentioned in section 4, we could firstly adapt an architecture model, and then, reflect the changes on the real system according to the difference operation between the adapted model and the current one. Therefore, it seems to be possible to deal with similar techniques in the field of *models@runtime* [16] combining model transformations with formal tools like Maude [17]. Finally, if the performance is satisfactory, we could consider formal tools as the final ***adaptation engine***.

Maude for Prototyping Self-Adaptation

The first benefit of using Maude for prototyping self-adaptive systems stems from its capability to provide designers with ***executable mathematical models*** of these systems. This capability becomes essential for adjusting and validating their adaptation behavior. Specifically, Maude can assist designers in (1) adjusting the abstraction level of context variables and variation points in the VML model with regards to the architecture details (e.g., the integers provided by the light sensor are translated into a boolean context variable to indicate whether there is or not light enough). The designer needs to decide how many values are enough to characterize a context variable or a variation point; (2) adjusting the design-time functions that specify the VML properties to be optimized; and (3) establishing the right links between the architectural elements and the VML variables (e.g., selecting thresholds to establish what means *false* or *true* in a boolean context variable). Regarding the abstraction level of variables in VML, it is worth noting that the number of considered alternatives has an impact on the adaptation stability (e.g., the more context values, the more potential situations to manage and, consequently, it may cause continuous and inefficient system reconfigurations), the computational cost (e.g., the more variants to handle the higher the cost) and the effectiveness of the process (e.g., an insufficient number of alternatives makes adaptation useless).

Concerning the ***main limitations*** we found when implementing the robotic example, Maude seems not to offer a simple method for solving constrained optimization problems. The current approach relies on exploring all the possible combination of variants to finally decide the best one in terms of the adaptation properties. Thus, the practical feasibility of this problem critically depends on the number of variants. Another barrier for the developers is the difficulty for debugging Maude programs, due to the concurrent nature of its rules and the scarcely legible traces it returns during the execution. We figured out that this difficulty increases exponentially as the number of rules grows, since it becomes more and more complex to follow the interactions between the Maude rules, causing undesired states in the system.

Automatic Code Generation from VML

We have identified some common Maude structures (partly described in section 4) that could be easily generated from a VML model (e.g., the context and the variability representation). However, part of the Maude specification cannot be generated from the VML model, as it also depends on the component-based architecture model, in particular of its reconfigurable and monitoring parts (related with the VML variation points and context variables, respectively).

For the future, we plan to continue exploring the potentials of Maude, in particular, for verifying the completeness and correctness of the self-adaptive behavior specifications. We also plan to link this work with our previous experience with VML [7] and with the MDE approach proposed by DiVA [6].

Acknowledgements and Additional Material. Juan F. Inglés-Romero thanks Fundación Séneca-CARM for a research grant (Exp. 15561/FPI/10). The authors would like to thank Prof. Antonio Vallecillo and Mr. Javier Troya for their helpful insights about modeling with Maude.

Additional material related to this work can be found in the following website: `https://sites.google.com/site/cvicentechicote/home/publications/varis2013`

References

1. Cheng, B.H.C., et al.: Software engineering for self-adaptive systems: A research roadmap. In: Cheng, B.H.C., de Lemos, R., Giese, H., Inverardi, P., Magee, J. (eds.) Self-Adaptive Systems. LNCS, vol. 5525, pp. 1–26. Springer, Heidelberg (2009)
2. Salehie, M., Tahvildari, L.: Self-adaptive software: Landscape and research challenges. ACM Transactions on Autonomous and Adaptive Systems 4(2), 1–42 (2009)
3. Maude website, `http://maude.cs.uiuc.edu/`
4. Inglés-Romero, J.F., et al.: Using Models@Runtime for Designing Adaptive Robotics Software: an Experience Report. In: 1st Int'l Workshop on Model-Based Engineering for Robotics (RoSym), Oslo, Norway, October 3-8 (2010)
5. Inglés-Romero, J.F., et al.: Towards the Automatic Generation of Self-Adaptive Robotics Software: An Experience Report. In: 20th IEEE Int'l Conf. on Collaboration Technologies and Infrastructures (WETICE), Paris, France, June 27-29, pp. 79–86 (2011)
6. EU 7FP DiVA Project, `http://www.ict-diva.eu`
7. Inglés-Romero, J.F., et al.: Dealing with Run-Time Variability in Service Robotics: Towards a DSL for Non-Functional Properties. In: 3rd Int'l Workshop on Domain-Specific Languages and Models for Robotic Systems (DSLRob 2012), held in conjunction with (SIMPAR 2012), Tsukuba, Japan, November 5 (2012)
8. The E-puck website, `http://www.e-puck.org`
9. Xtext website, `http://www.eclipse.org/Xtext/`
10. Hallsteinsen, S., Hinchey, M., Park, S., Schmid, K.: Dynamic software product lines. Computer 41(4), 93–95 (2008)
11. Fleurey, F., Solberg, A.: A Domain Specific Modeling Language Supporting Specification, Simulation and Execution of Dynamic Adaptive Systems. In: Schürr, A., Selic, B. (eds.) MODELS 2009. LNCS, vol. 5795, pp. 606–621. Springer, Heidelberg (2009)
12. The DiaSuite Project website, `https://diasuite.inria.fr/`
13. The MUSIC Project website, `http://ist-music.berlios.de/site/index.html`
14. MOSKitt4SPL website, `http://www.pros.upv.es/m4spl/index.html`
15. Muller, P.-A., Fleurey, F., Jézéquel, J.-M.: Weaving executability into object-oriented meta-languages. In: Briand, L.C., Williams, C. (eds.) MoDELS 2005. LNCS, vol. 3713, pp. 264–278. Springer, Heidelberg (2005)
16. Blair, G.S., Bencomo, N., France, R.B.: Models@run.time. IEEE Computer 42(10), 22–27 (2009)
17. Romero, J.R., Rivera, J.E., Durán, F., Vallecillo, A.: Formal and tool support for model driven engineering with Maude. Journal of Object Technology 6(9) (2007)

A Model Driven Software Product Line Process for Developing Applications

Sami Ouali[1], Naoufel Kraïem[1], Zuhoor Al-Khanjari[2], and Youcef Baghdadi[2]

[1] RIADI Lab, ENSI, Compus of Manouba, Tunisia
samiouali@gmail.com, naoufel.kraiem@ensi.rnu.tn
[2] Department of CS, Sultan Qaboos University, Oman
{zuhoor,ybaghdadi}@squ.edu.om

Abstract. Software Product Line Engineering (SPLE) is an approach for software reuse. It concerns to produce customized software products as atomic or composite services to be reused in SOA-based applications. A common set of artifacts is used to build these services in a planned and managed way. The main purpose of SPLE is to explore commonalities and variabilities. The SPLE approach provides a strategic software reuse that can produce quality Software as a Service (SaaS) while cutting cost and reducing time-to-market. This paper proposes a process to construct services as Software Product Lines by using Model Driven techniques. The process combines the use of maps, visual techniques for SPL modeling, especially features diagrams and MD techniques. In addition to the process, we have developed a tool to support map, feature, and class diagrams modeling.

Keywords: Model Driven, Software Product Line, Variability, Process, Tool.

1 Introduction

(SPLE) is an approach for software reuse. It is recognized as a successful approach to reuse in software development [1, 2]. It concerns with producing customized software products as atomic or composite services to be reused in SOA-based applications [23, 24]. SPLE allows companies to realize significant improvements on productivity. The purpose of this approach exploits the commonalities between products while preserving the ability to vary the functionality between these products. The difference between products is their variability, a key success in product lines [3]. SPLE distinguishes two layered levels of engineering [4]: Domain Engineering (DE) and Application Engineering (AE). DE deals with the identification of commonalities and variabilities among products. AE is used to develop a new product from a PL and the results from DE level are used to derive a specific product.

In the proposed process, we focus on both levels of engineering. First, we build the SPL at DE level. Then, we derive a specific product. The problem that we try to tackle in this paper is the discordance goal between the developed products and the real needs of its users.

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 447–454, 2013.

This paper presents a meta-model and a prototyping tool for requirement and feature modeling. The tool implements various models to assist stakeholders in the process of product configuration for SPLs. This paper describes an Eclipse plug-in for map, feature, and class modeling to model SPL, which assists SPL designer in the construction of SPL and the derivation of particular product. Providing tool as an Eclipse plug-in would facilitate the integration of these kinds of modeling with a development environment. The tool, itself, is built by using two Eclipse plugins: EMF, and GMF which provide many possibilities through extensions points.

The remainder of this paper is organized as follows. Section 2 summarizes a meta-model for intention and feature modeling in the context of SPLs. The proposed Model Driven for SPL process is described in section 3. The supporting tool is presented in section 4. Section 5 presents some related work. Finally, a conclusion section highlights our contribution and presents some research perspectives.

2 A Meta-model for SPL

This section describes a meta-model that synthesizes the interesting points such as SPL, intention, features, etc. We choose to transform this meta-model into a UML profile to (i) facilitate the integration with UML models, and (ii) use it in our MD approach. We try throw this meta-model to facilitate the relation between requirements model (MAP), features model and classes diagram. This meta-model is used in our process and in the generation of our tool support.

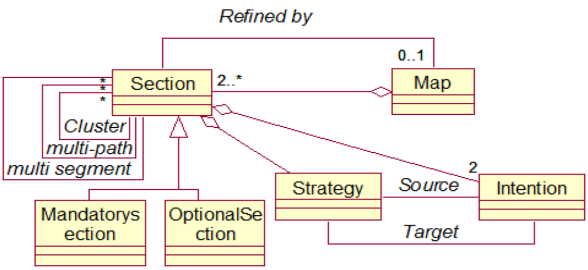


Fig. 1. Modification applied on map meta-model

A MAP [5] involves two or more sections. Each section is composed of two intentions and one strategy (these concepts are presented in section 3.1). A relationship between sections can be a cluster, multi-path or multi segment. To further use the meta-model in our process, we make a modification, as shown in Figure 1, by adding two types of sections: ‘OptionalSection’ and ‘MandatorySection’. These new sections are added for the mapping between map and features model.

A PL contains features, as shown in Figure 2. A product belongs to one PL. It is also composed of features that check some constraints (an exclusion and required relation) through the conflict and require relationships. Figure 2 presents the different concepts of feature Model. The possible relationships between features are presented by FeatureGroup. There are three relationships between features: 'And', 'Xor', and 'Or'. The relationships between a parent feature and its child features are categorized as: 'Mandatory' or 'Optional'.

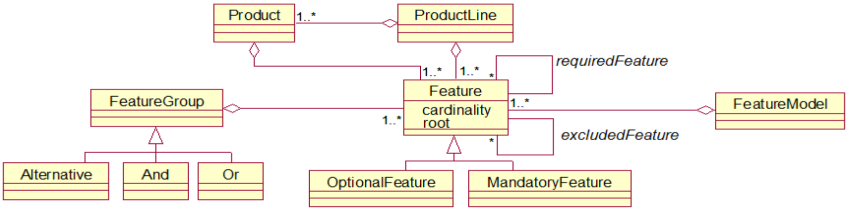


Fig. 2. Relationship between product line and features

Figure 3 presents the relation between the features elaborated in the analysis domain and the components constructed in implementation domain; the goal of implementation domain is to develop the components participating in the architecture of the SPL). A feature is associated to a component. There are two kinds of components: composite or leaf, i.e. simple component. A leaf is a non decomposable component. It has one or more possible implementation. A composite component is decomposed into subcomponents.

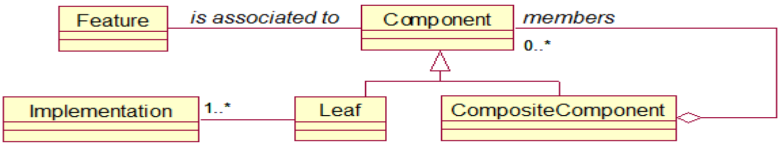


Fig. 3. Relationship between feature and component

Another alternative to the implementation of assets with components is the use of services as shown in Figure 4. There are two kinds of services: atomic service and composite service. An atomic service cannot further be decomposed. A composite service is composed of other services. A feature can be associated to a service.

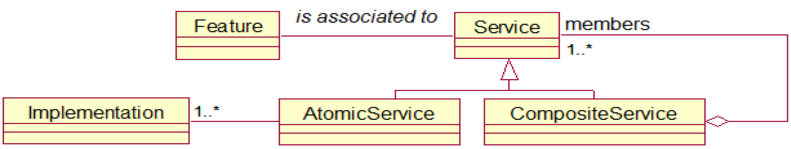


Fig. 4. Relationship between feature and service

3 Model-Driven SPL Process

This section presents the notations used in the model-driven SPL process. These are map model and features model. It also presents some rules and patterns for mapping between map, features, and class model. Both DE and AE are covered in the proposed as shown in Figure 5. The DE process deals with the creation of core assets. These core assets are PL requirements, analysis models, PL architecture, reusable components and services. This process concerns with the elicitation of intentions and strategies using the map for the design of user's requirements. In the modeling of the requirements, SPL features intentionality is represented as maps [22]. Then, features models derive class diagrams in the design view that used further used to obtain a component architecture or service architecture throw through components or services repository. The component architecture view deals with the creation of the interface type and a component. The interfaces contain the signature of operations. The generated component implements the required and provided interfaces. Service architecture view is also realized by generation of services from design view. The AE process deals with product derivation from the composition of existing artifacts created in the DE (existing components or services present in the components repository or services repository). It exploits the variability of the PL, which allows the satisfaction of users needs.

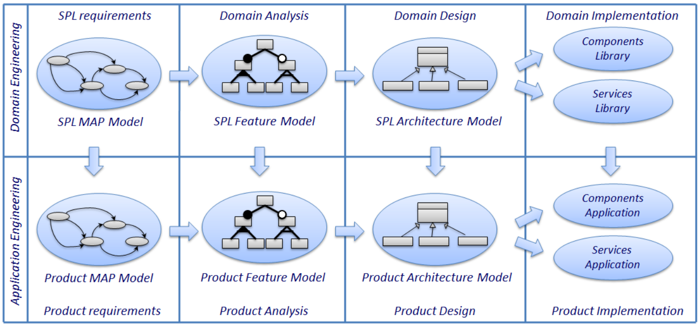


Fig. 5. Approach for SPL construction

3.1 MAP Model

A MAP [5] is a process model expressed in a goal-driven perspective that can provide~~s~~ a representation of a multi-facetted purpose based on a non-deterministic intentions and strategies. The directed nature of the graph shows which goals can follow which one. MAP is considered as intention-oriented process modeling that follows the human intention of achieving a goal [12]. A map is a directed graph from 'Start' to 'Stop', as the strategy shows the flow from the source to the target intention. It is composed of several sections. Intentions are represented as nodes of the graph. Strategies are the relationships between intentions. The key concepts of map are intention, strategy and section. An intention is a goal [6] that expresses a state that is expected to be reached or maintained. A strategy is a way or a means to achieve an intention. A section is a combination of two intentions linked by a strategy.

3.2 Feature Model

Feature modeling is a domain analysis technique, part of the Feature Oriented Domain Analysis (FODA) method [7], for developing software for reuse. This method has been applied in many domains such as telecom systems [8, 9], network protocols [10], and embedded systems [11]. Using feature modeling can help in generating domain design in SPL. Feature Models allow a representation of all possible products in a SPL, in terms of features. It represents the description of the mandatory features that are present in all the products of the PL along with variant features or optional features that do not appear in all the products.

In a feature model, we can find hierarchical relationships between features. The relationships between a parent feature and its sub features are categorized as mandatory features that are required in all the products, or optional features. A child feature can only appear in the products where its parent feature appears. Relationships between features are mandatory, optional, Xor, and-relationship and or. A feature~~s~~ model defines constraints which are compatibility rules. These rules refer to some restrictions in features combinations. They are two types of constraints: requires and excludes.

3.3 Patterns to Automate Transformations

Our process focuses on DE and AE, using goals and features model to establish architecture model. In this architectural model, we deal only with structural view (class diagram). We define some rules to try to automatically derive class diagram from requirement model. However, we recognize that the transformations that we propose cannot be univocal. We distinguish different kinds of transformations: (i) SPL requirements are mapped into feature models, and (ii) the structural information of the feature models is mapped into class diagrams. These transformations are implemented using QVT. We begin with the mapping between requirement model and analysis model, i.e. from MAP model to feature model, by using some transformations rules:

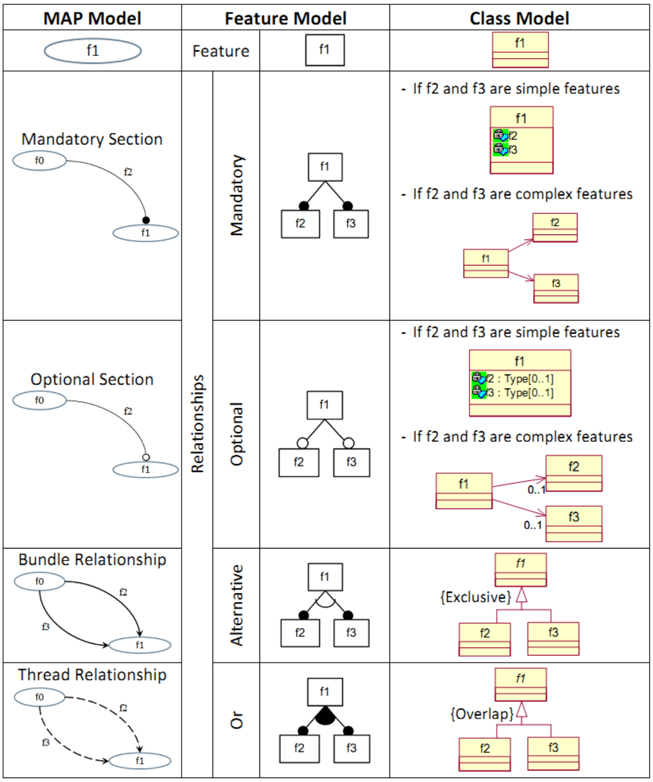
- A thread relationship in a map model representing a AND/OR relation between two strategies is equivalent to an Or-features-group
- A bundle relationship representing a XOR relation between two strategies is equivalent to an Alternative-features-group
- For mandatory and optional features, we have defined new concepts in the MAP meta-model: mandatory and optional section. The same graphical representation in the feature model is used for representing mandatory and optional sections.

Feature model are mapped into classes and attributes as shown in Table 1:

- A feature in the model tree can be simple or complex feature.
- The leaves of the tree can be simple features, in this case~~s~~ a simple feature is mapped into an attribute of a parent class.
- A non-atomic feature is mapped into a class representing this feature.
- Mandatory features imply a unidirectional association with the right cardinality.
- The optional features are mapped as associations with 0..1 as cardinality.

For alternative and OR groups of features, generalization concept is applied, where an alternative group of features is represented by a generalization of classes with exclusive as constraint between the subclasses, whereas Or group is represented by a generalization of classes with overlap as constraint between subclasses.

Table 1. Rules to automate transformations



4 Supporting Tool

Based on the meta-model presented in section 2, we developed an interactive, visual, tool for requirement modeling, feature capture, and architecture modeling. The tool will be extended to Feature Configuration. Our plug-in implements map modeling, cardinality-based feature modeling through transformations and architectures of SPL construction. This tool is based on eclipse plug-ins. The tool uses the multiple extension possibilities of offered by eclipse platform through two plug-ins: Eclipse Modeling Framework (EMF), and Graphical Modeling Framework (GMF).

These plugins provide many possibilities through extensions points, which can reduce the development effort. These plugins helps in constructing our model-driven approach. The proposed tool is built-by re-using these two Eclipse plugins. Providing a supporting tool for map, feature, and class modeling as an Eclipse plug-in is of paramount importance for with the integration of these kinds of modeling as part of a development environment. Object Constraint Language is used to describe constraints, rules and specifications. OCL statements can be specified as part of EMF Model or invoked directly from programming language (java).

5 Related Work

Several authors have proposed approaches relating to feature models and product architectures. The mapping between requirements and design is a complex task because of the flexibility and adaptability of the SPL, the different technology possibilities, etc. Van Lamsweerde [13] derives software architectures from the formal specifications of a system goal model. Bragança et al. [20] propose to obtain features from use case models. Laguna et al. [21] proposes some patterns to obtain design view and use case models from features models.

Many proposals express variability with UML models. [18] propose the extension of UML Meta model by the adding of new relationships "option" and "alternative". Clauß proposes the use of stereotypes to express variability [19].

Many propositions are concerned with the construction of editor for features modeling. AmiEddi [14] supports feature modeling notation [4] and didn't support feature and group cardinalities. CaptainFeature [15] implements a cardinality-based notation. Some other works try to integrate feature-based configuration like ConfigEditor [11].

We also found some commercial tools (Pure::Variants [16] or GEARS [17]). Pure::Variants support feature modeling and configuration by using a tree-view render without feature cardinalities. It allows modeling of constraints between features and uses Prolog-based constraint solver for the configuration. GEAR allows the modeling and configuration of software variants.

6 Conclusion and Future Work

There is growing interest on the topic of SPL construction to derive particular product. We have aimed at fulfilling stakeholder's requirement of the SPL and the particular requirements. We have focused on the mapping of goal-oriented requirements into software architectures. First, we have defined a Meta model to represent the different concepts related to the construction of SPL. Then, we have defined a process to (i) construct SPL, (ii) derive particular product throw configuration and model derivation. We have emphasized the construction of services for SOA-based applications. The process generates design views by using class diagram derived from a requirement model. The process is supported by rules and mapping patterns and realized by a tool using EMF and GMF plugins to facilitate the construction of SPL.

Our future works includes (i) the formal validation of the proposed mapping patterns and (ii) the generation of assets of the SPL. These assets will be PL requirements, analysis models, PL architecture, reusable components, and services.

References

1. Clements, P., Northrop, L.: Software Product Lines: Practices and Patterns. Addison-Wesley, Boston (2001)
2. Bosch, J.: Design & Use of Software Architectures, Adopting and Evolving a product-line approach. Addison-Wesley (2000)

3. Van Grup, J.: Variability in Software Systems, the key to software reuse (Thesis). University of Groningem, Sweden (2000)
4. Czarnecki, K., Eisenecker, W.: Generative Programming: Methods, Tools, and Applications. Addison-Wesley (2000)
5. Rolland, C., Prakash, N., Benjamen, A.: A Multi-Model View of Process Modeling. Requirements Engineering Journal 4(4), 169–187 (1999)
6. Jackson, M.: Software Requirements & Specifications – a Lexicon of Practice, Principles and Prejudices. ACM Press, Addison-Wesley (1995)
7. Kang, K., Cohen, S., Hess, J., Nowak, W., Peterson, S.: Feature-oriented domain analysis (FODA) feasibility study. Technical Report CMU/SEI-90TR -21, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA (1990)
8. Griss, M., Favaro, J., Alessandro, M.: Integrating feature modeling with the RSEB. In: Proceedings of the Fifth International Conference on Software Reuse (1998)
9. Lee, K., Kang, K.C., Lee, J.J.: Concepts and guidelines of feature modeling for product line software engineering. In: Gacek, C. (ed.) ICSR 2002. LNCS, vol. 2319, pp. 62–77. Springer, Heidelberg (2002)
10. Barbeau, M., Bordeleau, F.: A protocol stack development tool using generative programming. In: Batory, D., Blum, A., Taha, W. (eds.) GPCE 2002. LNCS, vol. 2487, p. 93. Springer, Heidelberg (2002)
11. Czarnecki, K., Bednasch, T., Unger, P., Eisenecker, U.: Generative programming for embedded software: An industrial experience report. In: Batory, D., Consel, C., Taha, W. (eds.) GPCE 2002. LNCS, vol. 2487, pp. 156–172. Springer, Heidelberg (2002)
12. Soffer, P., Rolland, C.: Combining Intention-Oriented and State-Based Process Modeling. In: Delcambre, L.M.L., Kop, C., Mayr, H.C., Mylopoulos, J., Pastor, Ó. (eds.) ER 2005. LNCS, vol. 3716, pp. 47–62. Springer, Heidelberg (2005)
13. van Lamsweerde, A.: From system goals to software architecture. In: Bernardo, M., Inverardi, P. (eds.) SFM 2003. LNCS, vol. 2804, pp. 25–43. Springer, Heidelberg (2003)
14. Selbig, M.: A feature diagram editor: analysis, design, and implementation of its core functionality. Kaiserslautern, Germany (2000)
15. Bednasch, T., Endler, C., Lang, M.: Captain Feature (2002-2004) Tool available on Source Forge at, `https://sourceforge.net/projects/captainfeature/`
16. Pure-systems GmbH. Variant Management with Pure:Consul. Technical White Paper (2003), `http://web.pure-systems.com`
17. Krueger, C.W.: Software mass customization. White paper, `http://www.biglever.com/papers/BigLeverMassCustomization.pdf`
18. Von der Massen, T., Lichter, H.: RequiLine: A Requirements Engineering Tool for Software product lines. In: Software Product-Family Engineering, Siena, Italy (2003)
19. Clauß, M.: Generic modeling using Uml extensions for variability. In: Workshop on Domain Specific Visual Languages at OOPSLA (2001)
20. Bragança, A., Machado, R.J.: Automating Mappings between Use Case Diagrams and Feature Models for Software Product Lines. In: Proceedings of SPLC, Kyoto, Japan (2007)
21. Laguna, M.A., González, B.: Product Line Requirements: Multi-Paradigm Variability Models. In: Proceedings of the 11th Workshop on Requirements Engineering, Spain (2008)
22. Ouali, S., Kraiem, N.: From Intentions to Software Design using an Intentional Software Product Line Meta-Model. In: The Proceeding of the 8th International Conference on Innovations In Information Technology. Al Ain, UAE (2012)
23. Al Rawahi, N., Baghdadi, Y.: Approaches to Identify and Develop Web Services as Instance of SOA Architecture. In: Proc. of the IEEE-ICSSSM 2005, China (2005)
24. Baghdadi, Y.: A Methodology for Web services-based SOA realization. International Journal of Business Information Systems 10(3), 264–297 (2012)

Privacy in the Cloud: Bridging the Gap between Design and Implementation

Vassilis Manousakis[1], Christos Kalloniatis[1], Evangelia Kavakli[1], and Stefanos Gritzalis[2]

[1] Cultural Informatics Laboratory, Department of Cultural Technology and Communication, University of the Aegean, University Hill, GR 81100 Mytilene, Greece
{kavakli,ct08081}@ct.aegean.gr, chkallon@aegean.gr

[2] Information and Communication Systems Security Laboratory, Department of Information and Communications Systems Engineering, University of the Aegean, GR 83200, Samos, Greece
sgritz@aegean.gr

Abstract. Bridging the gap between design and implementation stages has been a major concern that deplores designers, analysts and developers for quite a long time during the design and implementation of information systems in traditional environments. This issue grows to bigger dimension with the presence of cloud computing. Designing and modeling an Information System for the Cloud is a major and hard task that most of the traditional software engineering approaches fail to fulfill. In parallel, many respective organisations and respective researchers have highlighted a number of security and privacy challenges that are not present in traditional environments and need special attention when implementing or migrating information systems into a cloud environment. Thus, security and privacy are by themselves two areas that need special attention in the cloud era. This paper moves on to this direction. Specifically, it presents a number of privacy-oriented technical concepts that analysts need to consider when designing and modeling privacy-aware systems in a cloud environment. Also it suggest for every concept a number of implementation techniques that can assist developers in implementing the respective concepts.

Keywords: Cloud Computing, privacy concepts, CSA, Implementation Techniques, Software Engineering, PETs.

1 Introduction

Cloud Computing is without a doubt one of the most significant innovations presented in the global technological map. The number of potential users enrolling and using cloud services increases exponentially on a daily bases. The great demand from online users for cloud services along with the reduced operational costs that the latter offers has motivated many organisations and companies to consider implementing from scratch or migrating organizational services, data and applications on the Cloud. However, despite the various positive characteristics of all cloud service models like reduced costs, better availability insurance, on demand data storage and computation

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 455–465, 2013.

power, cloud users have expressed major concerns regarding the protection of their privacy in such environments basically due to the distributed character of the cloud architecture and the involvement of different stakeholders and providers on specific applications and data processing mechanisms.

According to National Institue of Standards and Technology (NIST), Cloud Computing delivers three different types of services to the end users that derive from three different models. The delivery models are IaaS, PaaS and SaaS, each one of them providing three distinct types of resources, like virtual infrastructure resources, application platforms and software services. Each delivery model is considered as separate layer that is depended from each other and with IaaS being the foundation, PaaS sits on top of IaaS and SaaS sits on top of PaaS. So, as the end users combine different type of services, capabilities form each layer are inherited, so as privacy issues. Also another factor that should be considered is the impact of deployment model on privacy. Privacy risks seem to have bigger impact on public, hybrid and community cloud, compared to the other deployment models. On the other hand, cloud consumers should have in mind that despite the fact that private and cloud deployments are theoretically safer, but still the same threats apply and the only thing that changes is the users group. In this deployment model, the users from the administrator to the simple user are trusted, but that does not mean that proper measures should be considered.

In general, the more low level services the client requests the more responsible for security and privacy is, but still the cloud vendor has an important role on managing and implementing security and privacy measures even in low levels of abstraction.

The scope of the paper is twofold. Firstly, as far as we know, it makes one initial step on identifying and describing the major privacy-related concepts that are newly introduced into the cloud. Secondly, it aims on bridging the gap between design and implementation stages by suggesting for each privacy concept a number of implementation techniques for realizing these concepts on a cloud environment. Specifically the paper is structured as follows. Section 2 describes in text and graphically the privacy related concepts. In section 3 the respective implementation techniques are presented that realize the aforementioned concepts. Finally, section 4 concludes the paper and suggests future extensions.

2 Privacy-Oriented Concepts

In order to preserve privacy inside the cloud, certain requirements need to be realised. This section describes the basic privacy properties that constitute the basic issues that need to be considered when designing or migrating to the cloud. Specifically, the aim of this section is twofold. Firstly, it aims on revealing and describing a number of privacy related concepts derived from related literature as well as respective cloud threats identified so far both in text and diagrammatically.

Secondly, it aims on identifying the applicability of every concept on the respective cloud service model thus assisting the stakeholders on deciding which privacy properties need to be realised in order to satisfy their own goals on every cloud service model respectively. The concepts proposed are mainly derived from the European Commission Draft Report on Security Issues in Cloud Computing [5] as well as from our previous work presented in [6-13]. However, new concepts are also introduced and explained in order to form a complete set for covering all the respective cases. For every concept a brief description is described along with the main privacy issue and the main threats existing from the respective literature regarding this issue.

2.1 Isolation

The specific concept is referred to the complete seal of user's data inside the Cloud computing environment. Isolation is meant to address data disclosure in two ways, firstly, from purpose limitation point of view and secondly from the aspect of the proper technical implementation techniques [5]. Cloud computing resources are shared among a multi-tenant environment. Thus, excessive cloud employee's access rights, posing the risk of any kind of Personal Identifiable Information disclosure and thus violating client's privacy. The specific concept is matched with the following threats derived from [1], Abuse and Nefarious Use of Cloud Computing, Insecure interfaces and APIs, Malicious Insiders, Shared technology issues, Data Loss or Leakage, Privileged user access and Lack of Data Segregation.

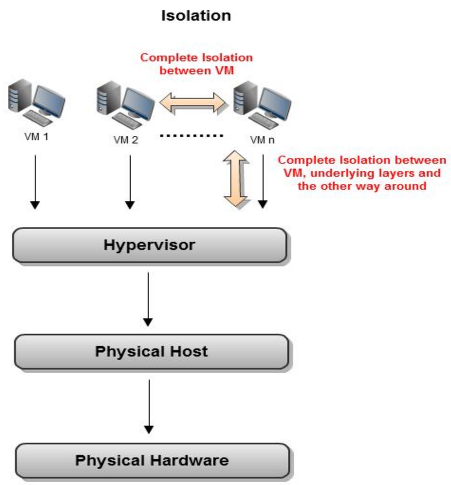


Fig. 1. Isolation Example

2.2 Provenanceability

The specific concept is referred to the provenance of the data related to the authenticity or identification, the quality of the results of certain procedures, modifications, updates and vulnerabilities, the provenance of certain actions inside the cloud, the detection of origins of security violations of an entity[14], the auditability of client's data and matters that are related to the cloud's sub-system geographical dispersion referred to the legal issues, regulations, policies and each country's rules as far as data processing and protection is concerned. All the above constitute a potential privacy violation if they are not realised properly by implementing the appropriate technical measures. The specific requirement is matched with the CSA threats, Malicious Insiders, Privileged user access, Regulatory Compliance, Data Location, Investigate Support.

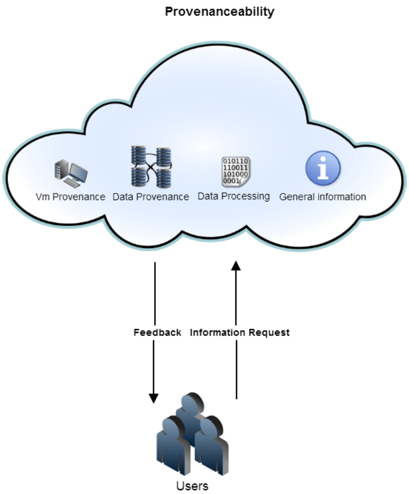


Fig. 2. Provenaceability Example

2.3 Traceability

Traceability concept aims to give the user the ability, to trace her data or not. This property is examined from the proper/improper data erasure aspect, which is a major problem in web-based systems and still continues to exist in clouds. Many cases have been documented for privacy violation due to improper data deletion (documents, photos, etc.). The traceability concept aims to protect privacy, through the ability of tracing them among the data repositories and reassuring that the data have been completely deleted or maintained invisible and anonymized after their deletion[1]. The clients should be able to trace the physical location of their data and to be able to verify that they are processed according to their collection purpose. The specific concept is matched with the CSA threats, Malicious Insiders, Data Loss or Leakage, Regulatory Compliance, Data Location.

---

[1] In some cases, certain cloud service providers apply retention policies as far as data are concerned. That means that for several reasons, that are stated inside the contract between the cloud provider and the client, the data remain at rest after the clients deletion request for some time and are strictly accessed form specific personnel and only for certain purposes.

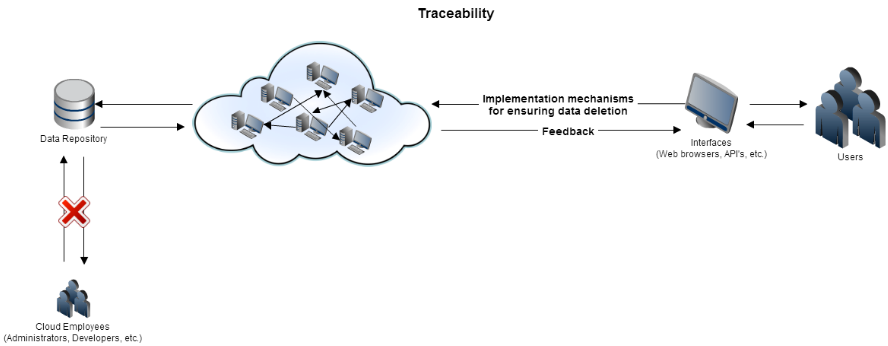


Fig. 3. Traceability Example

2.4 Intervenability

Intervenability concept is referred to the fact that, the users should be able to have access and process their data despite the cloud's service architecture. A cloud vendor may rely on other provider's subcontractor services in order to offer her services. That should not be an obstacle for the user to intervene[2] with her data in case she suspects that her privacy is violated by the subcontractors. In fact cloud vendor must be able to provide all the technical, organizational and contractual means for accomplishing this functionality for the user including all respective subcontractors that the vendor cooperates and interrelates [5]. The same applies for the situation that a cloud vendor or the subcontractors are bankrupted and client's data are moved to another provider. The specific concept is matched with the CSA threats, Unknown Risk Profile, Data Location.

2.5 Accountability

Accountability concept is referred to the fact that cloud providers should be able to provide at any given time information about their data protection policies and procedures or specific cloud incidents related to users' data. The cloud architecture[3] makes a complex form of an information system. In terms of management and audit controls, this fact could result in very difficult manageability of the protections mechanisms and incidents. In case of a privacy violation, a cloud provider should be able in any given time to provide information about what, when and how an entity acted and which procedures followed to tackle it [5]. The specific concept is matched with the CSA threats, Abuse and Nefarious Use of Cloud Computing, Insecure interfaces and APIs, Malicious Insiders, Shared technology issues, Data Loss or Leakage, Account or Service Hijacking Unknown Risk Profile, Privileged user access, Regulatory Compliance, Data Location, Lack of Data Segregation, Lack of Recovery, Investigate Support, Long-term Viability.

---

[2] Access, rectification, erasure, blocking and objection.

[3] International services residual.

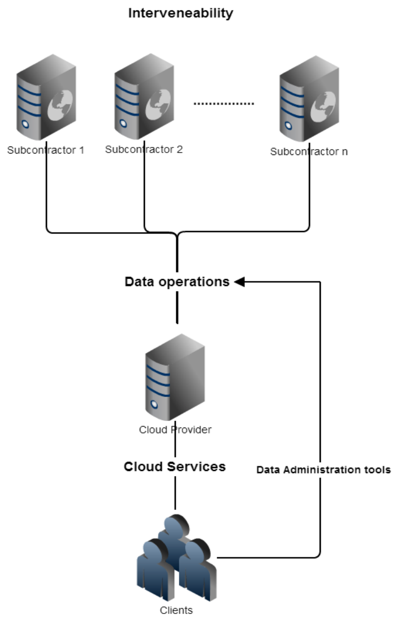


Fig. 4. Intervenability Example

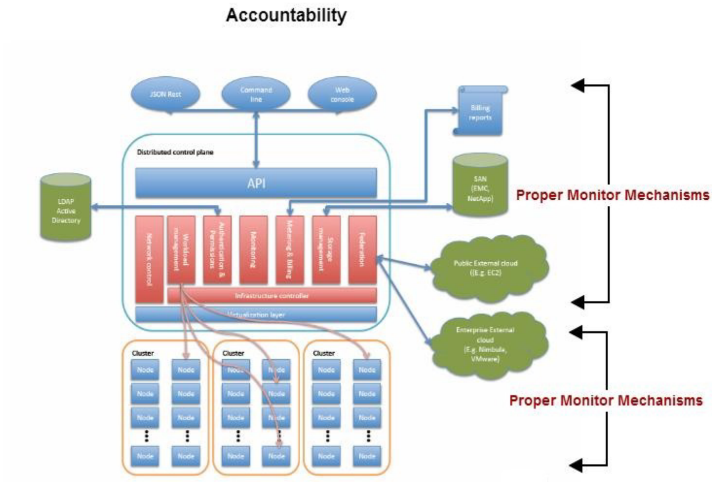


Fig. 5. Accountability Example

Table 1. Matching Security and Privacy Properties with Cloud Services Models

	IaaS	SaaS	PaaS
Property #1: Isolation	X	X	X
Property #2: Provencability	X		
Property #3: Traceability		X	
Property #4: Intervenability	X	X	X
Property #5: Accountability	X	X	X

In table 1 a matching between the aforementioned concepts and the cloud service models is presented. Based on the aforementioned table analysts can identify which the privacy concepts are, that belong to their system, and how these concepts can constitute an initial obstacle during the design of the information system on a cloud environment. Usually when analysts consider the cloud deployment scenario their main goal is to decide on which service model they are interested in deploying to. The identified concepts and the respective matching is a start for creating a holistic process for assisting analysts on receiving the proper decisions.

3 Implementation Techniques

3.1 Data Filtering Techniques (Firewalls)

A firewall is a security guard that is placed between an internal[4] and an external environment. The functions that constitute this mechanism on a simple form are two, data filtering and acceptance or rejection of incoming and outgoing packets. In our case privacy preservation is ensured through the implementation of filtering techniques that aim to achieve isolation between two virtual machines inside a virtual network, through the analysis and detection of malicious traffic that is sent to and from a virtual machine (vm) through the router. Recent editions of firewalls are implementing intrusion detection and prevention inside their core functions, which is a pro in privacy preservation.

3.2 Encryption Mechanism

Encryption mechanisms are used in order to ensure the secrecy of important information [11] inside the cloud environment. Encryption techniques are implemented in various areas of the cloud, in order to encrypt data flow[5] or data at rest[6] and thus protecting privacy through ensuring strong isolation and anonymization of sensitive data [16, 17].

---

[4] The environment that needs to be protected from the external environment.
[5] Virtual and physical networks.
[6] Databases.

3.3 Hypervisor Hardening, Language, Sandbox, Virtual Machine, OS – Kernel, and Hardware Based Isolation

All the above mentioned implementation techniques provide logical isolation between different entities, procedures and operations inside the cloud. Two types of isolation are implemented, logical and hardware based isolation. Logical isolation is achieved from the first five techniques and attempts to seal all the procedures, operations and the data that flow through the installation of multiple isolation layers between cloud parts, with different programming techniques, inside the cloud environment. On the other hand, hardware based isolation is achieved through hardware controls and it's provided by the processors or by special components combined with the processor [18].

3.4 Privacy Policies and Contracts

Appropriate privacy policies and contracts that benefit client's interest as far as privacy protection is concerned. Cloud users must be very careful about the terms and conditions of the service they are using in order to ensure that their privacy is not violated in case of an incident or a situation that needs to be cleared, e.g. data hosting in foreign countries, what happens in case the cloud provider is bankrupted etc. [5, 7, 17].

3.5 Forensics

Forensics mechanisms are essential in case of an incident, in order to be determined under what circumstances the incident occurred and who is responsible. For example in cloud computing is important to know the origins of the processed data or the detection of fault and security and privacy violations provenance [1, 14, 15, 17].

3.6 Identity Management (IdM)

In this category fall technologies that the use of them combined or individually protect the client's privacy through a solid identity system. Certain techniques that constitute this category are biometrics, smart cards, permission management components, etc. All the above mentioned are techniques that can protect privacy through a solid isolated virtual system and detecting the provenance of certain actions and propably prevent them because of the proper defined identity inside the system [2,6].

3.7 Data Tracking

Data tracking techniques are referred to the technologies that enable data tracing processes in order to inform the client about the route path that their data have followed, where they are hosted at the current time and in what state they have been. Privacy is ensured through the fulfillment of the provencability and traceability requirement that detect the provenance of the data and the provide information about whether client's data are deleted or not and where are located [2, 17].

3.8 Process Operation Identification and Validation

The identification and validation of the processes that modify data is essential to whether the outcomes are reliable or and ensure that privacy is not violated by malicious processes [17].

3.9 Privacy Preserving Data Mining

Most of the times service providers are using collected data from the users, e.g. data traffic, search history, configurations, in order to examine them and make a customer profile for marketing purposes. The fact that personal data are examined is considered as a privacy violation if it's not done properly. This kind of procedures should provide basic anonymization through the data analysis in order for the client's privacy to be ensured [19].

3.10 Monitor and Auditing

Monitor and auditing techniques are widely known and used in order to preserve privacy through monitoring and auditing functions and procedures that occur to an informational system. Monitor and audit procedures are incorporated into security tools and help protecting client's privacy and provide information as to who is accountable about something inside the cloud environment [10].

Table 2. Matching Security and Privacy Properties with Implementation techniques

	Data Filtering (Firewalls)	Encryption	Hypervisor Hardening	Language based Isolation	Sandbox Isolation	Virtual Machine Isolation	Os – Kernel based isolation	Hardware – based Isolation	Privacy Policies and Contracts	Forensics	Identity Management (IdM)	Data Tracking Techniques	Process operations identification and validation	Privacy Preserving Data Mining	Monitor and Auditing
Property #1: Isolation	x	x	x	x	x	x	x	x			x				
Property #2: Provencability									x	x	x	x	x		
Property #3: Traceability		x							x			x			
Property #4: Intervenability									x						
Property #5: Accountability	x	x	x							x	x			x	x

The techniques described above, fall in the category of privacy-enhancing technologies since their main focus is on realizing privacy related concepts as the ones identified in this work. However, these technologies focus on the software implementation alone, irrespective of the privacy issues as well as the cloud services on which the respective software system will be based upon. Thus, this matching aims on providing an initial step of how to bridge the gap between the main privacy concerns and the respective technologies used specifically for cloud environments.

On the other hand, security and privacy requirements methodologies, which address early stages of system design, focus on privacy-related organisational requirements, but do not link these requirements to implementation solutions. Following a number of concepts for understanding the relationship between the user needs in the organisational domain and the capabilities of the supporting software systems is of critical importance and this paper takes an initial step to this direction.

4 Conclusions

The various innovations that cloud computing introduced in its operational environment vary from the traditional "trusted" environment where today's information systems rely on. These innovations hinder new privacy concepts that need to be identified in order to protect the design and implementation of new information systems or even for traditional systems when migrating on cloud environments. Based on this, the specific paper presents an initial effort on identifying the basic privacy-oriented concepts that need to be considered when designing information systems for the cloud. Also, it moves one step further by bridging the gap between design and implementation phases by suggesting a number of privacy-enhancing technologies specifically for the cloud environments. The contribution of this paper can be adopted by a traditional privacy requirements engineering approach in order to be enhanced with the respective concepts aiming on the realization of an approach that deals with the design of cloud oriented systems as it is conducted with the traditional ones respectively. This is also the main future extension for our work. Specifically future steps include the transformation of these concepts on technical requirements and the design of a modeling process for applying these requirements on a real case scenario.

References

1. Cloud Security Alliance, Top Threats to Cloud Computing V1.0, `https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf` (retrieved September 22, 2012)
2. Heiser, J., Nicolett, M.: Assessing the Security Risks of Cloud Computing, white paper, Gartner group, ID Number: G00157782 (June 3, 2008)
3. Subashini, S., Kavitha, V.: A survey on security issues in service delivery models of cloud computing. Journal of Network and Computer Applications 34(1), 1–11 (2010)
4. Cloud Security Alliance, Security Guidance for Critical Areas of Focus in Cloud Computing V3.0, `https://cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf`(retrieved September 22, 2012)
5. Draft, EU Directive for Security issues in Cloud Computing (2012)
6. Kalloniatis, C., Kavakli, E., Gritzalis, S.: PriS Methodology: Incorporating Privacy Requirements into the System Design Process. In: Mylopoulos, J., Spafford, G. (eds.) Proceedings of the SREIS 2005 13th IEEE International Requirements Engineering Conference – Symposium on Requirements Engineering for Information Security. IEEE CPS, Paris (2005)

7. Kalloniatis, C., Kavakli, E., Gritzalis, S.: Addressing privacy requirements in system design: The PriS method. Requirements Engineering 13(3), 241–255 (2008)
8. Kalloniatis, C., Kavakli, E., Kontellis, E.: PRIS tool: A case tool for privacy-oriented Requirements Engineering. Journal of Information Systems Security 6(1), 3–19 (2010)
9. Kavakli, E., Kalloniatis, C., Loucopoulos, P., Gritzalis, S.: Incorporating Privacy Requirements into the System Design Process: The PriS Conceptual Framework. Internet Research, Special issue on Privacy and Anonymity in the Digital Era: Theory, Technologies and Practice 16(2), 140–158 (2006)
10. Kalloniats, C., Kavakli, E., Gritzalis, S.: Dealing with Privacy Issues during the System Design Process. In: 5th IEEE International Symposium on Signal Processing and Information Technology, Athens, Greece, December 18-21, pp. 18–21 (2005)
11. Mouratidis, H., Kalloniatis, C., Islam, S., Huget, M.P., Gritzalis, S.: Aligning Security and Privacy to support the development of Secure Information Systems. Journal of Universal Computer Science (2012)
12. Mouratidis, H., Giorgini, P.: Secure Tropos: A Security-Oriented Extension Of The Tropos Methodology. International Journal of Software Engineering and Knowledge Engineering (2007)
13. Mouratidis, Giorgini, P.: Security Attack Testing (SAT) - testing the security of information systems at design time. Inf. Syst. 32(8), 1166–1183 (2007)
14. Wei, L., et al.: Managing Security of Virtual Machine Images in a Cloud Environment (2009)
15. Microsoft Technical report: Privacy in the cloud computing era, a Microsoft perspective, Microsoft Corp, Redmond, USA (November 2009)
16. Sonehara, N., et al.: Isolation in Cloud Computing and Privacy – Enhancing Technologies – Suitability of Privacy – Enchancing Technologies for Separating Data Usage in Business Processes, National Institute of Informatics, Chiyoda-Ku, Tokyo (2005)
17. Zhang, O.Q., et al.: How To Track Your Data: The Case for Cloud Computing Provenance, HP Laboratories, HPL-2012-11 (2012)
18. Viswanathan, A., Neuman, B.C.: A survey of isolation techniques. Draft Copy, University of Southern California
19. Singh, M.D., et al.: A cryptography based privacy preserving solution to mine cloud a data, Infosys Technologies Limited, Bangalore, India (2010)

A Scalable Multi-Party Protocol for Privacy-Preserving Equality Test

Maryam Sepehri, Stelvio Cimato, and Ernesto Damiani

Dipartimento di Informatica, Università Degli Studi di Milano
Crema, Italy
{firstname.lastname}@unimi.it

Abstract. Multi-party computation (MPC) is attractive for data owners who are interested in collaborating to execute queries without sharing their data. Since data owners in MPC do not trust each other, finding a secure protocol for *privacy-preserving* query processing is a major requirement for real world applications. This paper deals with *equality test* query among data of multiple data owners without revealing anyone's private data to others. In order to nicely scale with large size data, we show how communication and computation costs can be reduced via a *bucketization* technique. Our bucketization requires the use of a trusted third party (TTP) only at the beginning of the protocol execution. Experimental tests on horizontally distributed data show the effectiveness of our approach.

Keywords: secure multi-party computation, equality test query, privacy-preserving query processing and bucketization.

1 Introduction

In today's collaborative environments there is an increasing need of performing computations over data held by multiple owners without sharing the data themselves. This requirement is the subject of an important area of research, called *privacy-preserving* computation. An important selection of privacy-preserving computations consists in executing queries over partitioned databases while keeping privacy. For instance, *equality test* queries [1, 2] selecting items in a database that are equal to a given value, or *range* queries [1–3], selecting values in a fixed range. In partitioned databases, owners can be required to execute even more complex operations like *intersection*, which computes the set of items common to all the private databases. Secure multi-party functions are often represented as combinational circuits [4, 5]. This solution is quite efficient in two-party case but does not scale to the multi-party case. Goldreich et al. [5] showed that a multi-party version of Yao's protocol needs a quadratic communication complexity in the number of parties. Over the years, the research community has developed a wide range of privacy-preserving techniques to realize different functions across the shared databases based on *homomorphic encryption*, *oblivious polynomial evaluation* and *commutative encryption* methods [6–10]. Homomorphic encryption has been deployed in [10] where a solution for two-party equality

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 466–477, 2013.

has been presented. Assuming that the receiver and the sender have a private set of s elements, the computation and communication complexity amounts to $O(2s^2 lgN)$, where N is the RSA-modulus. The homomorphic encryption method is not straightforwardly extendable to multi-party case because if multi-party equality test runs in pairs, each party will learn the intersection with the remaining ones. In [11], the authors present a solution based on oblivious polynomial evaluation (OPE) protocol. The protocol requires each party to perform s oblivious evaluations of a polynomial of degree n with communication cost $O(n)$. For this reason, this protocol is considered too expensive to implement in the multi-party setting. Li and Wu [12] propose a protocol that involves a TTP to compare the values held by two other parties using homomorphic encryption. Although this protocol is faster than OPE, it has two main drawbacks: 1) the TTP should be trusted by all parties; 2) when the number of parties increases, the solution does not scale because of the communication and computation bottleneck created at TTP. The set intersection problem is to compute the intersection between these sets without revealing any other information. Freedman, Nissim and Pinkas [9] propose a protocol based on the Paillier's cryptosystem mentioned above. According to [13], the average computation cost for each party and the communication cost among all parties are $O(s(s+m)lgN)$ and $10s(m-1)^2 lgN$, respectively. Following Freedman's protocol, Kissner and Song [8] designed a technique for set intersection problem using polynomial representations with $O(cs^2 lgN)$ computation complexity and $2cm(5s+2)lgN$ communication complexity, where c is a suitable constant. Sang et al. [13] adopt a distinct through related approach with lower computation and communication costs with respect to [9, 8] with $O(csmlgN)$ computation cost for each party and $2cm(4s+5)lgN$ communication cost. A major step forward was taken by Agrawal et al. [7], who introduce a two-party protocol for set intersection based on commutative encryption with linear complexity. The protocol has been extended to multi-party case [14] with lower complexity than [9, 8, 13], i.e., $O(smlgN)$ computation and $((m-1)s+(m-2)s+1+(m-1))lgN$ communication overhead. However, Vaidya's [14] extension only computes the size of the intersection, without identifying the intersection's elements. Most of these techniques offer strong privacy guarantees, but they do not scale well for large databases because of using heavyweight cryptographic operations or several rounds of interactions among parties.

A central issue in this context is designing a protocol for the **S**ecure **M**ulti-party **E**quality test **P**roblem (SMEP), which is secure and efficient when the number of parties and the size of data increases. In this paper we present a new protocol, B-SMEQ (**B**ucketized **S**ecure **M**ulti-party protocol for **E**quality test **Q**ueries) to address the SMEP problem, which adopts a *bucketization* technique to reduce the time complexity. Our solution uses a commutative encryption scheme to avoid information being revealed among data owners. In order to make the protocol fast, we divide data into buckets so that work can be done considering only a subset of data. To realize the bucketization scheme, TTP is involved in an initial phase. However, the TTP does not participate in the query processing, avoiding the creation of computation and communication bottleneck.

2 Problem Statement

We consider a Multi-party System$MS = \langle D, m, U, q, R\rangle$ for equality test computation involving five different entities as illustrated in Fig. 1: a common database D with n records, which has been horizontally partitioned among a set of distrustful data owners O in which $|O| = m$; a set of users U allowed to search; an equality test query q and the result R of the query computation across multiple partitions. The data owners $O = \{O_1, ..., O_m\}$ hold data partitions $\{T_1, ..., T_m\}$, respectively and we assume that all partitions include a searchable attribute A with a set of values V_A, and another attribute B with a set of values V_B. Moreover, for every $i \in \{1, ..., m\}$ we denote with $T_{i,A}$ the column corresponding to attribute A, and for each $v \in T_{i,A}$, with $ext_i(v)$ each value occurring in V_B where $T_{i,A} = v$.

In SMEP, the m data owners must jointly compute the result R to the equality test query q and return it to an authorized user $u \in U$ without revealing their private data to each other, satisfying: *data privacy* property, since the user should learn just the result of the query; the *query privacy* property, since the data owners should not learn the query; and *query anonymous result,* since the user should not know whom the results belong to. In our model, the data owners are honest but curious, meaning that they follow the protocol as exactly specified, but they try to learn extra information by analyzing the transcript of messages received during the execution. Owners are arranged in a ring topology communicating each other via secure channels; one data owner is designated as the master site (*initiator*)[1].

Fig. 1. Basic Scenario

2.1 Preliminaries

Commutative Encryption Scheme. Informally, a commutative encryption is a pair of encryption functions f and g such that $f(g(v))=g(f(v))$. By using the combination $f(g(v))$ to encrypt v, we can ensure that no data owner can perform encryption without the help of other data owners. For a more formal definition of commutative encryption scheme, we remand the reader to [7].

Set Intersection Protocol. The set intersection protocol proposed by Agrawal et al. [7] runs between two parties S(sender) and R(receiver), holding a set of values V_S, V_R and keys k_s, k_r, respectively.

[1] Since user authentication and access control are not the main focus in this paper, we assume the authorization between the data owners and users are appropriately managed.

Step 1. Both S and R apply hash function h to their own values and encrypt the result with their secret keys, $f_{k_s}(h(V_S))$ and $f_{k_r}(h(V_R))$.
Step 2. Sites S and R exchange $f_{k_s}(h(V_S))$ and $f_{k_r}(h(V_R))$ after randomly reordering their values to prevent possible inference attacks.
Step 3. Site S encrypts each value of the set $f_{k_r}(h(V_R))$ with k_s to obtain $Z_R = f_{k_s}(f_{k_r}(h(V_R)))$ and sends back the pairs ($f_{k_r}(h(V_R)), Z_R$) to R.
Step 4. Site R creates pairs (v,Z_R) by replacing $f_{k_r}(h(V_R))$ with the corresponding v where $v \in V_R$.
Step 5. Site R selects all v for which $Z_R \in f_{k_r}(f_{k_s}(h(V_S)))$ for $V_S \cap V_R$.

3 A Protocol for Equality Test

In this section, we extend Agrawal's protocol for solving the SMEP problem. Then, we introduce an improvement to ensure scalability. In the basic version, called SMEQ, we extend set intersection protocol from two-party to multi-party setting, where the role of sender is played by each of m data owners. SMEQ is simple to describe and implement but suffers from high communication and computation. We address these issues in an improved version protocol, B-SMEQ, which makes use of bucketization technique on the searchable attribute.

3.1 Protocol 1:SMEQ

Input. A multi-party system (see Section 2) with the data owner O_1 as initiator[2].
Output. The result of user query q to partitions $\{T_1, \ldots, T_m\}$ where $[(T_{1,A} = v) \vee \ldots \vee (T_{m,A} = v)]$.

Step 1. Both user (receiver side, R) and data owners O (sender side, S) apply a hash function h to the value v and to the set of their values of attribute A, respectively. Let $v' = h(v)$ be the result of hashing from the receiver side and let $T'_{i,A} = h(T_{i,A})$, for each $i \in \{1, \ldots, m\}$, be the hashing of the set values $T_{i,A}$. Sites S and R randomly choose a secret key, k_r for R and $\langle k_i, k'_i \rangle$ for data owner O_i.
Step 2. R spans the encrypted hash value y_R=$f_{k_r}(v')$ to all data owners at site S.
Step 3. At site S, each data owner O_i, $1 \leq i \leq m$, does the following:
3.1 Computes $f_{k_i}\left(T'_{i,A}\right) = Y_i = \{y_i = f_{k_i}(x) | x \in T'_{i,A}\}$
3.2 Generates a set of new keys, one for each value of attribute B, as K_i^B=$\{k_{ix} = f_{k'_i}(x) | x \in T'_{i,A}\}$
3.3 Encrypts each value x in $T_{i,B}$ with the corresponding key k_{ix} to obtain $Y_i^B = \{E_{k_{ix}}(u) | u \in T_{i,B}\}$ where E is an encryption function, which can be efficiently inverted given key (k_{ix})

[2] Any data owner (for instance, the one holding the largest data partition) can be chosen as initiator.

3.4 Computes $I_i = f_{k'_i}(y_R)$ for the purpose of decrypting the values of attribute B at site R

3.5 Owner O_i randomly reorders the tuples Y_i and Y_i^B and sends them along with I_i to the next owner $O_{(i \ mod \ m)+1}$

3.6 Data owner $O_{(i \ mod \ m)+1}$ encrypts only Y_i with the key $k_{(i \ mod \ m)+1}$ and sends the triple $\langle f_{k_{(i \ mod \ m)+1}}(Y_i), Y_i^B, I_i \rangle$ after reordering to the next participant in the ring.

This process is repeated until Y_i is encrypted by all keys of m data owners, obtaining $Z_i = f_{k_1}(f_{k_2}(...(f_{k_m}(Y_i))))$, up to a permutation of the encryption keys[3]

Step 4. Each data owner O_i sends $\langle Z_i, Y_i^B, I_i \rangle$ to owner O_1.

Step 5. Owner O_1 receives all tuples from Step 4 in order to initiate a two-party Agrawal's set intersection protocol (see Section 2.1) between the user as a receiver site and the initiator as a sender site.

Step 6. Owner O_1 passes y_R through the ring in order to have it encrypted by all keys $k_1, \ldots, k_m$ for obtaining $y'_R = f_{k_1}(\ldots(f_{k_m}(f_{k_r}(v^{'}))))$, and then sends back y'_R together with $\langle Z_i, Y_i^B, I_i \rangle$ to the user, for all $i \in \{1, \ldots, m\}$.

Step 7. First, R decrypts $y_R^{'}$ with her decryption key to obtain $y_R^{''} = f_{k_1}(\ldots(f_{k_m}(v^{'})))$, and then for each i, $1 \leq i \leq m$: 1) Finds tuples in Z_i whose entry related to attribute A is equal to $y_R^{''}$; 2) Considers the entry corresponding to attribute B of those tuples; 3) Decrypts I_i with k_r, obtaining $f_{k'_i}(v')$; 4) Uses $f_{k'_i}(v')$ to decrypt the corresponding entry in Y_i^B.

Although this protocol is simple and effective, it has high cost in terms of communication and computation over large data sets (see Section 5). Moreover, it suffers from high query computation workload at user side. In the next section, we use data bucketization to improve the protocol.

3.2 Protocol 2: B-SMEQ

In this Section, we describe the protocol B-SMEQ, which adopts a bucketization technique on searchable attribute to reduce the communication and computation costs, while preserving the user and owner privacy and the result anonymity. B-SMEQ makes use of a TTP that is not involved in the query processing, but only in the realization of the bucketization scheme. Before the protocol starts, the TTP builds an interchange matrix W(see Phase 1- Step 3) and sends the row vectors of W to the data owners. The owner that receives W_1 is called initiator. Then, data owners are arranged into a ring; each holds a permutation[4] of the bucket order $\{1, \ldots, s\}$ and a vector from matrix W. In the first part of the protocol, each owner sends her buckets in encrypted form to the next participant. When a user wants to submit an equality query, she gets from the TTP the bucket number of the query value according to the TTP's own permutation, $\bar{\Pi}$.

[3] The keys $k_1, k_2, \ldots, k_m$ represent a commutative set of keys.

[4] For the purpose of this paper, we assume that partitions are abundant meaning that the number of permutations is much greater than the number of participants.

This number is sent to the initiator who uses it to identify the bucket related to the user value from her predecessor in the ring she should overencrypt with her key. The procedure is repeated for each node of the ring. In the second phase of the protocol, selected buckets (i.e., the ones corresponding to the user value) are propagated along the ring until they have been encrypted by all keys. Once all data owners hold the encrypted buckets, a two-party Agrawal's protocol is executed between the initiator and the user. In the next sections we will provide a detailed description of the algorithm.

Defining Buckets. We divide the range of values, in the domain of each searchable attribute A, into buckets so that each bucket is assigned a unique label (ID). This label is then stored alongside the encrypted version of the searchable attribute. We adopt equi-width strategy for selecting buckets, which divides the range of possible values of searchable domain into s buckets of the same size l, i.e. $l = \frac{A_{max}-A_{min}}{s}$, where A_{max} and A_{min} are the maximum and minimum values in the domain of A, respectively. Thus, we define bucket boundaries of the searchable attribute as $BU = \{B_1 : [A_{min},\ l],\ \ldots,\ B_s : (l(s-1),\ A_{max}]\}$. BU is called *public bucketization scheme* and it is accessible to all data owners and authorized users as well.

The Protocol. B-SMEQ has two phases: *computation of matrix* W and *query protocol*. First, in order to preserve data privacy, each data owner O_i separately computes a local permutation $(B_{\pi_{i_1}},\ \ldots B_{\pi_{i_s}})$ of the public bucketization scheme for the searchable attribute. Each data owner i chooses a local permutation $\Pi_i = (\pi_{i_1}, \ldots, \pi_{i_s})$ of bucket indices $(1,\ 2,\ \ldots,\ s)$. Moreover, the TTP chooses her own permutation $\bar{\Pi}$ and sends it to the user posing the query.

Let's now assume that the user query value v falls into bucket B_i of public bucketization. Then, the user sends to a pre-set data owner (henceforth, the *initiator*) a query for the bucket $B_{\bar{\pi}_i}$. In this way, when data circulate along the ring, data owners will not know which bucket ID the user is looking for. We will now present our matrix-based mechanism, which allows each data owner to correctly select the local bucket corresponding to $B_{\bar{\pi}_i}$.

Phase 1. Computation of Matrix *W*

Step 1. Each data owner O_i sends her permutation Π_i to the TTP.
Step 2. The TTP builds the matrix Π containing the received permutation vectors and generates a $m \times s$ interchange matrix W, where the matrix elements are defined by Eq. (1).

$$\Pi = \begin{pmatrix} \Pi_1 \\ \Pi_2 \\ \vdots \\ \Pi_m \end{pmatrix} = \begin{pmatrix} \pi_{11} & \pi_{12} & \ldots & \pi_{1s} \\ \pi_{21} & \pi_{22} & \ldots & \pi_{2s} \\ \vdots & & & \vdots \\ \pi_{m1} & & & \pi_{ms} \end{pmatrix} W = \begin{pmatrix} W_1 \\ W_2 \\ \vdots \\ W_m \end{pmatrix} = \begin{pmatrix} w_{11} & w_{12} & \ldots & w_{1s} \\ w_{21} & w_{22} & \ldots & w_{2s} \\ \vdots & & & \vdots \\ w_{m1} & & & w_{ms} \end{pmatrix}$$

In the following equation, we denote by $\bar{\Pi}^{-1}(l)$ the position in vector $\bar{\Pi}$ that contains value l: $w_{ij} = \pi_{i\delta_i}\ \forall i \in \{1, 2, \ldots, m\},\ j \in \{1, 2, \ldots, s\}$ where

$$
\begin{aligned}
&\delta_1 = \bar{\Pi}^{-1}(j) \\
&\delta_2 = \bar{\Pi}^{-1}(k_1) \quad \text{with } k_1 \text{ s.t. } w_{1k_1} = j \\
&\vdots \\
&\delta_i = \bar{\Pi}^{-1}(k_{i-1}) \\
&\qquad \text{with } k_{i-1} \text{ s.t. } w_{1k_{i-1}} = k_{i-2}, \\
&\qquad \text{with } k_{i-2} \text{ s.t. } w_{2k_{i-2}} = k_{i-3}, \\
&\qquad \vdots \\
&\qquad \text{with } k_1 \text{ s.t. } w_{i-1k_1} = j
\end{aligned} \tag{1}
$$

The rationale behind Eq. (1) is generating the entries of matrix W by identifying a corresponding entry of matrix Π. This entry, for the first row ($i = 1$) is obtained by looking for the position of index j in the TTP permutation. For the other rows , the entry is obtained by looking for the position of index j in the previous rows of matrix W itself. For instance to compute δ_2 for $w_{21} = \pi_{2\delta_2}$, TTP follows this procedure: (a) Find $j = 1$ in the previous line of matrix W, (b) Take the position of 1 and read the value in this position of the TTP permutation.

Step 3. The TTP sends the row vectors of W to the data owners and her permutation $\bar{\Pi}$ to the user.

Phase 2. Query Protocol

Steps 1-3 of B-SMEQ correspond to Steps 1-3 (excluding Step 3.5) of SMEQ

Step 4. The user sends $B_{\bar{\pi}_k}$ to the initiator, where k is the number of the bucket where the query value v falls.

Step 5. Let us recall that, at this step, each data owner O_i holds data Y_{i-1} (which corresponds to T_{i-1} encrypted with the key k_{i-1}) of O_{i-1}. With this arrangement, O_2 is the algorithm's initiator[5]. O_2 sets $h_2 = w_{1\bar{\pi}_k}$, selects $Y_1(h_2)$ i.e. the bucket in Y_1 whose ID is h_2, and encrypts it with her own key. Then O_2 sends h_2 to the next owner. When data owner O_i receives h_{i-1} from O_{i-1}, he sets $h_i = w_{(i-1)h_{i-1}}$, selects the corresponding bucket $Y_{i-1}(h_i)$ and sends the position h_i to the next owner. This step iterates until all data owners have selected their bucket.

Step 6. The data owners apply the procedure described in Step 3.5 of SMEQ just on the selected buckets obtained at Step 5.

Steps 7-10 of B-SMEQ correspond to the Steps 4-7 of SMEQ

4 Correctness and Privacy Issues

Correctness. In order to show that B-SMEQ computes query results correctly, we will sketch a simple proof showing that data owners always select the position corresponding to the bucket ID including the user value (Step 5).

[5] This is not a limitation, because any node can be selected as initiator by changing the order of permutations at the time of W generation by TTP.

- Owner O_2 who has the data Y_1 of O_1 receives $\bar{\pi}_a$, computes $h_2 = w_{1\bar{\pi}_a}$, selects $Y_1(h_2)$ and sends h_2 to the O_3. Observe that by definition $h_2 = w_{1\bar{\pi}_a} = \pi_{1\delta_1}$, where $\delta_1 = \bar{\Pi}^{-1}(\bar{\pi}_a)$. Hence $h_2 = \pi_{1a}$, which means O_2 chooses the correct bucket.
- Owner O_i who has data Y_{i-1} of O_{i-1} receives h_{i-1} from party $i-1$, computes $h_i = w_{(i-1)h_{i-1}}$, selects $Y_{i-1}(h_i)$, and sends h_i to next owner. Recall that by Eq. (1), we have $h_i = w_{(i-1)h_{i-1}} = \pi_{(i-1)\delta_{i-1}}$, where $\delta_{i-1} = \bar{\Pi}^{-1}(k_{i-2})$ and

$$\begin{aligned} &k_{i-2} \text{ s.t. } w_{1k_{i-2}} = k_{i-3} \\ &\vdots \\ &k_2 \text{ s.t. } w_{(i-3)k_2} = k_1 \\ &k_1 \text{ s.t. } w_{(i-2)k_1} = h_{i-1} \end{aligned} \tag{2}$$

Since by definition $h_{i-1} = w_{(i-2)h_{i-2}}$, we have that $w_{(i-2)k_1} = w_{(i-2)h_{i-2}}$, hence $k_1 = h_{i-2}$. Again by definition, $h_{i-2} = w_{(i-3)h_{i-3}}$ and then $w_{(i-3)k_2} = w_{(i-3)h_{i-3}}$ that implies $k_2 = h_{i-3}$. By iterating this procedure, we obtain $k_{i-3} = h_2$, that implies $w_{1k_{i-2}} = h_2 = w_{1\bar{\pi}_a}$. This means $k_{i-2} = \bar{\pi}_a$, since $\delta_{i-1} = a$, proving the correctness for data owner O_i. □

Privacy of B-SMEQ. Here, we provide an informal analysis of B-SMEQ privacy. Here, we just consider processing a single query. A more detailed analysis including multiple queries will be carried out as future work. We focus on the privacy of Steps 5 and 6 where the records of each data owner are selected according to the user bucket ID and circulated in order to be encrypted by all keys.

- *Indistinguishability of data distribution*: At each round of the query protocol, every data owner receives a new set from her predecessor via the ring. Since each value of the received set has been hashed and encrypted using commutative encryption function. The distribution of encrypted hash values are indistinguishable from the uniform random distribution.
- *Protection from relation inference*: In each round of Step 6, data owner i receives a data set encrypted with different keys, so that party i can not infer the relationship between received data sets.
- *Protection from bucket inference*: In addition to the size of the whole upstream data set (revealed in the set intersection protocol), in Step 5 each party can learn the size (number of tuples) of the upstream selected bucket. Nevertheless, when a data owner O_i overencrypts the encrypted tuples received from her predecessor, owner O_i does not know which bucket ID corresponds to the received tuples. The size of the selected position does not reveal any further information to the data owner.

According to the above analysis, our protocol is secure against semi-honest parties (see Section 2) as long as no two data owners collude.

5 Time Complexity Analysis

5.1 Theoretical Cost Analysis

Computation Cost. The main computation cost during the execution of both protocols belongs to hashing and encrypting the set of values V_A corresponding

Table 1. Computation and Communication costs of SMEQ and B-SMEQ with m=10 data owners, s=5 buckets and t equal to 10% of the number of records

Number of records	C_{SMEQ}	C_{B-SMEQ}	C'_{SMEQ}	C'_{B-SMEQ}
50000	$660.022 \cdot 10^3$	$300.022 \cdot 10^3$	$5450.645 \cdot 10^3$	$991.145 \cdot 10^3$
100000	$1320.022 \cdot 10^3$	$600.022 \cdot 10^3$	$10900.645 \cdot 10^3$	$1981.145 \cdot 10^3$
200000	$2640.022 \cdot 10^3$	$1200.022 \cdot 10^3$	$26160.774 \cdot 10^3$	$4753.375 \cdot 10^3$
300000	$3960.022 \cdot 10^3$	$1800.022 \cdot 10^3$	$39240.774 \cdot 10^3$	$7129.374 \cdot 10^3$
500000	$6600.022 \cdot 10^3$	$3000.022 \cdot 10^3$	$65400.774 \cdot 10^3$	$11881.374 \cdot 10^3$

to n distributed records among m data owners[6]. Let C_h be the cost of evaluating the hash function h, C_f be the cost of encryption/decryption by function f, C_E be the cost of encryption/decryption by function$E$, and let t be the number of tuples satisfying the equality match[7]. Following the SMEQ protocol step-by-step, it is possible to quantify the complexity of each step and compute the overall complexity of the protocol. For the sake of conciseness, we omit the details, but report the total computation cost of SMEQ :

$$\begin{aligned} C_{SMEQ} &= (n+1) + 1 + (nm+n) + (n+m) + m + 2t \\ &= nm + 3n + 2m + 2 + 2t \in O(mn) \end{aligned} \tag{3}$$

As regards the B-SMEQ protocol, a reduction factor s where s is the number of buckets, is to be considered affecting the evaluation of the computation cost. Table 1 shows the computation costs C_{SMEQ} and C_{B-SMEQ} for the protocols SMEQ and B-SMEQ, respectively. We supposed t to be equal to 10 percent of the number of records. B-SMEQ has significantly lower computation cost for the record cardinality values of interest for practical applications.

Communication Cost. Communication cost can be computed as the total number of bits transmitted during the protocol execution. We compute the total communication cost of our protocols under the assumption that each data owner has $\frac{n}{m}$ expected records (uniform distribution). We consider k as the length of each encrypted codeword of the domain of encryption function f (see Section 2) and k' the length of the encryption function E on other attribute. The communication cost of SMEQ can be computed by looking each step of the procedure, for the sake of conciseness, here we only give the final result that is $O(mn)$:

$$C'_{SMEQ} = (m^2 + nm - \tfrac{n}{m} + 3m + n - 1) \cdot k + (nm - \tfrac{n}{m} + n) \cdot k' \tag{4}$$

The total communication cost of B-SMEQ is:

$$\begin{aligned} C'_{B-SMEQ} &= (m^2 + 3m + \tfrac{1}{s}(mn - \tfrac{n}{m}) + 2ms + \tfrac{n}{m} - 1) \cdot k \\ &+ (\tfrac{n}{m} + \tfrac{1}{s}(nm - n - \tfrac{n}{m})) \cdot k' \end{aligned} \tag{5}$$

Table 1 presents the communication costs C'_{SMEQ} and C'_{B-SMEQ} of SMEQ and B-SMEQ. We assumed k and k' to be equal to the smallest integer greater than

[6] We compute the total computation cost of the two mentioned protocols with the assumption that each data owner has $\frac{n}{m}$ expected records (uniform distribution).

[7] For the sake of simplicity in the calculations below, we do not consider the cost of reordering encrypted values and assume a unitary cost when applying C_h, C_f and C_E to a single record.

$\lg(n)$. As Table 1 shows, the B-SMEQ protocol has lower communication cost; also, in this range, the difference between the two protocol increases with the number of records.

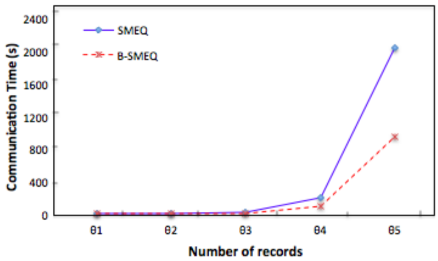
5.2 Practical Cost Analysis

In this section, we verify the scalability of our protocol via some experimental tests. We report the results in terms of communication time on a Linux machine with dual Intel CPU running at 2.26 GHz. and 2GB Ram. We use Castalia[8] simulator for Wireless Sensor Networks. In particular, for our simulation, we deploy 4 and 5 nodes, respectively for SMEQ and B-SMEQ. For both protocols, we create a ring of three nodes with numbers from 0 to 2 for the three data owners and one node (number 3) for the querier[9]. To encrypt the set of searchable attribute of each data owner's table, we implement a simple *commutative encryption* protocol based on *exponentiation modulo p*. Since the most communication time of the two protocols is devoted to exchanging data of each data owner along the ring in order to be encrypted by all keys, we only focus on Steps 3.4 and 3.5 of SMEQ, and Steps 5 and 6 of B-SMEQ. We run two different experiments: in the first one, we compare the two protocols; in the second one, we evaluate the effect the number of buckets has on the communication time for B-SMEQ.

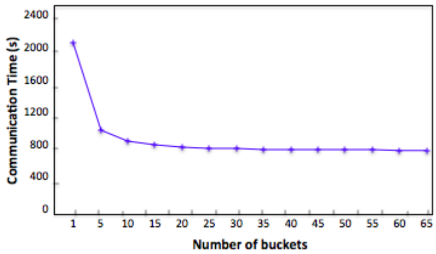
In the first experiment, we set five different triples $(\theta 1, \ldots, \theta 5)$ of number of records as $\theta 1 = \{5, 6, 7\}$, $\theta 2 = \{50, 60, 70\}$, $\theta 3 = \{500, 600, 700\}$, $\theta 4 = \{5000, 6000, 7000\}$ and $\theta 5 = \{50000, 60000, 70000\}$, where the position j in each triple is the number of records for data owner $j \in \{1, 2, 3\}$. Note that for B-SMEQ, we divide the searchable attribute domain $[1 \ldots 100]$ into s=5 number of buckets of the same size $l = 20$. Figure 2(a) shows the result of our simulation, the solid line displays the result from SMEQ, whereas the dotted line displays the result from B-SMEQ. The difference in communication time between SMEQ and B-SMEQ increases fairly slowly when the number of records of data owners is relatively small, but it grows much faster as the number of records increases. The results come from the fact that for each query, in SMEQ all records of data owners must pass through the ring, while in B-SMEQ only records corresponding to the bucket ID of user's query are taken into consideration. In the second simulation, we study the effect of increasing the number of buckets on communication time for B-SMEQ. We fix the number of records given by $\theta 5$ and searchable attribute with range $[1 \ldots 100]$ and repeat the experiment with s=5k, where $k \in \{1, \ldots, 13\}$. In Figure 2(b), the x axis shows the number of buckets and y axis shows the total communication time. Interestingly, with respect to varying the number of buckets in ascending order, we can see a progressive communication time decreasing. For instance, when the number of buckets is $s = 5$, B-SMEQ provides about 2 times improvement over SMEQ (i.e. s=1). The reason is that, increasing the number of buckets, the expected number of records "falling" in each bucket decreases. Moreover, the improvement due to bucketization is higher for $1 \leq s \leq 10$, since when $s > 10$

[8] `http://www.omnetpp.org/component/content/article/8-news/3478`

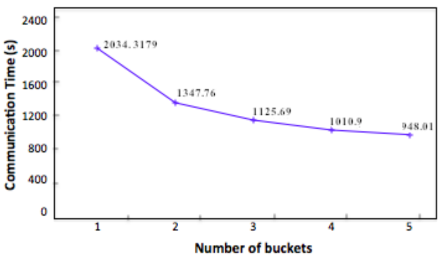
[9] It should be noted that for B-SMEQ we need to deploy an initiator node for the role of TTP.



(a)



(b)



(c)

Fig. 2. (a) Comparison of SMEQ and B-SMEQ Protocols based on communication time (m=3, s=5), (b) Effect of increasing number of buckets on communication time (θ_5, m=3, $1 \le s \le 65$) and (c) Effect of increasing number of buckets on communication time (θ_5, m=3, $1 \le s \le 5$)

the expected number of records in each bucket and for each data owner does not considerably change. In order to further clarify this concept, in Figure 3(c) we show the results of the same experiment when $s = \{1, 2, 3, 4, 5\}$. Our results show that bucketization decreases communication time dramatically at first; then, the marginal contribution of additional buckets to speed-up tends to decrease. This behavior suggests finding the optimal number of buckets, i.e., the number of buckets where the marginal contribution speed of the additional bucket is negligible. This behavior happens regardless of data distribution (uniform and normal).

6 Conclusions

In this paper, we proposed the B-SMEQ protocol to compute equality test queries in multi-party setting. Unlike existing approaches, our protocol scales well to large size data and it is designed to work with more than two parties. The protocol adopts a bucketization technique to reduce the workload and make the algorithm fast even when the number of records in the private databases considerably increases. Experimental tests on randomly generated databases with around $2{\cdot}10^5$ records confirm the efficiency on our protocol. Since each range query can be translated into a sequence of equality queries [15], our protocol can be straightforwardly extended to manage *range* queries. Moreover, an extension to join queries is underway, taking into account table fragmentation along multiple data owners.

References

1. Damiani, E., Vimercati, S.D.C., Jajodia, S., Paraboschi, S., Samarati, P.: Balancing confidentiality and efficiency in untrusted relational dbmss. In: Proceedings of the 10th ACM conference on Computer and communications security, CCS 2003, pp. 93–102. ACM, New York (2003)
2. Hacigümüş, H., Iyer, B., Li, C., Mehrotra, S.: Executing sql over encrypted data in the database-service-provider model. In: Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, SIGMOD 2002, pp. 216–227. ACM, New York (2002)
3. Boneh, D., Waters, B.: Conjunctive, subset, and range queries on encrypted data. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 535–554. Springer, Heidelberg (2007)
4. Yao, A.C.C.: How to generate and exchange secrets. In: Proceedings of the 27th Annual Symposium on Foundations of Computer Science, SFCS 1986, pp. 162–167. IEEE Computer Society, Washington, DC (1986)
5. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game. In: Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing, STOC 1987, pp. 218–229. ACM, New York (1987)
6. Agrawal, R., Asonov, D., Li, M.K.: Sovereign joins. In: Proceedings of the 22nd International Conference on Data Engineering (2006)
7. Agrawal, R., Evfimievski, A., Srikant, R.: Information sharing across private databases. In: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, SIGMOD 2003, pp. 86–97. ACM, New York (2003)
8. Kissner, L., Song, D.: Privacy-preserving set operations. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 241–257. Springer, Heidelberg (2005)
9. Freedman, M.J., Nissim, K., Pinkas, B.: Efficient private matching and set intersection, pp. 1–19. Springer (2004)
10. Lindell, Y., Pinkas, B.: Privacy preserving data mining. J. Cryptology 15(3), 177–206 (2002)
11. Naor, M., Pinkas, B.: Oblivious polynomial evaluation. SIAM J. Comput. 35(5), 1254–1281 (2006)
12. Li, R., Wu, C.: Co-operative private equality test. I. J. Network Security 1(3), 149–153 (2005)
13. Sang, Y., Shen, H., Tan, Y., Xiong, N.: Efficient protocols for privacy preserving matching against distributed datasets. In: Ning, P., Qing, S., Li, N. (eds.) ICICS 2006. LNCS, vol. 4307, pp. 210–227. Springer, Heidelberg (2006)
14. Vaidya, J., Clifton, C.: Secure set intersection cardinality with application to association rule mining. Journal of Computer Security 13(4), 593–622 (2005)
15. Damiani, E., De, S., Vimercati, C., Paraboschi, S., Samarati, P.: Computing range queries on obfuscated data. In: Proc. of the Information Processing and Management of Uncertainty in Knowledge-Based Systems (2004)

Goal-Oriented Security Requirements Analysis for a System Used in Several Different Activities

Haruhiko Kaiya[1], Takao Okubo[2], Nobuyuki Kanaya[2], Yuji Suzuki[1], Shinpei Ogata[1], Kenji Kaijiri[1], and Nobukazu Yoshioka[3]

[1] Dept. of Computer Science, Shinshu University, Nagano 380-8553, Japan
kaiya@shinshu-u.ac.jp
[2] Fujitsu Laboratory, Kawasaki, Kanagawa, 211–8588, Japan
[3] National Institue of Informatics (NII), Tokyo 101-8430, Japan

Abstract. Because an information system is used in different activities simultaneously today, we have to analyze usages of the system in the existing activities and to-be usages in an intended activity together. Especially, security aspects should be carefully analyzed because existing activities are not always secure. We propose a security requirements analysis method for resolving this problem. To take both existing and intended activities into account together, we integrate them on the basis of the unification of common actors. To explore possible attacks under integrated activities, we enumerate achievable attacks on the basis of the possible means in each actor with the help of security knowledge. To avoid or mitigate the attacks and to achieve fundamental goals, we disable some means or narrow down the means to be monitored with the help of propositional logic formulae. Through case studies on insurance business, we illustrated our idea.

Keywords: Goal-Oriented Requirements Analysis, Security Requirements Analysis, Strategic Dependency, Logic.

1 Introduction

When an information system is intended to be introduced into some activity such as business or amusements, we have to analyze the activity and define what kinds of functionalities and qualities including security needs are required for the system. In many cases, such a system or a part of it has already used in other activities. For example, we are using PC's, Web browsers, emails, social network services (SNS) and free software for our daily activities. Some business person cannot perform his/her business without own smart phones, and he/she also uses SNS for fun via the smart phones. We thus have to take such existing activities into account when an information system is intended to be introduced into an activity. Especially, we have to take them into account carefully for analyzing security requirements because not all existing activities are secure.

In this paper, we propose a method for analyzing security requirements for an information system used in several different activities. When actors and their dependencies in each activity are specified, we can clarify the common parts among different activities. Through such common parts, good or bad impacts are conveyed. We thus use i* notation [1,2] for specifying each activity. In i*, the needs of actors are represented as goals. Although some goals are fundamental for some actor and others are just subcontracts, there is no explicit distinction between fundamental goals and others in i*. We

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 478–489, 2013.

thus introduce this distinction because fundamental goals cannot be easily modified or given up but others can. Instead, we don't use original four goals types in i*. In usual goal-oriented modeling, a goal hierarchy is represented in a goal tree or a goal DAG (directed acyclic graph) so that lower goals satisfy their upper goals. In i*, such a goal hierarchy is used for specifying the rationale of each actor, and a proper actor will have his/her rationale so that lower goals satisfy their upper goals. However, this does not go for malicious actors. For example, some free software or web site sometimes contains a virus against its official functionalities. We thus introduce the concept "incorrect goal decomposition". To explore avoiding or mitigating security problems, we simply use propositional logic formulae because the rationale for each actor is represented in goal hierarchies and each of them can be represented in a propositional logic formula. Although existing i* variations could be used for analyzing security requirements for a system used in several activities, our method is simpler than others and even practitioners can accept it.

The rest of this paper is organized as follows. In the next section, we review related works including i* modeling language. In section 3, we explain our method in detail. In section 4, we apply our method to an insurance business to evaluate our method. Finally, we conclude our current results and show the future issues.

2 Related Work

A modeling language i* [1,2] is one of well-known goal-oriented requirements notations, and there already exists a lot of its security extensions [3], [4] [5], [6], [7]. When we develop and deploy a new information system, several human, organizations and existing systems will be related the news system. In i*, such human, organizations, existing systems and the new system are represented as actors. Because each actor has its goals and not all goals can be achieved by itself, an actor depends on another. Such dependencies are represented as strategic dependencies in i*. The strategic dependency is the most novel idea in i*, and most of all its variations such as Tropos [8] contain the idea. In i*, four types of goals such as (hard) goals, soft-goals, tasks and resources are used. Although these types are useful for some purposes, distinguishing them correctly is not easy. Especially, tasks are very useful for specifying a means for achieving a goal, but the other usages are not always effective. Although a soft-goal is not the same as a quality requirement, soft-goals are usually used for representing quality requirements. Although quality requirements are used for specifying how well a function is established [9], the relationship between quality requirements and a functional requirement is not clear in i* SD (Strategic Dependency) model.

We review several security extensions of i*. In secure tropos by Mouratidis [4], security concerns can be attached to both want- and can-relations in i* strategic dependencies. These concerns can be used for developing strategic rationale model, i.e., goal hierarchy in each actor However, how to write such concerns is out of scope of this study. In Secure i* [6], [7], each actor is assumed as an attacker, and attacks are assumed on the basis of the means of each actor. However, assumed attacks are limited because each actor will belong to several different business or activities in the real life. There is another secure tropos by Giorgini et al. [3]. In their research, additional relationship types such as ownership, trust and delegation is used in addition to dependencies. They use logic based formal techniques for analyzing the model. Because the model becomes more complex than normal i*, it seems to be difficult to write a model in such extension. Sutcliffe also proposed an extension of i* using trust [5].

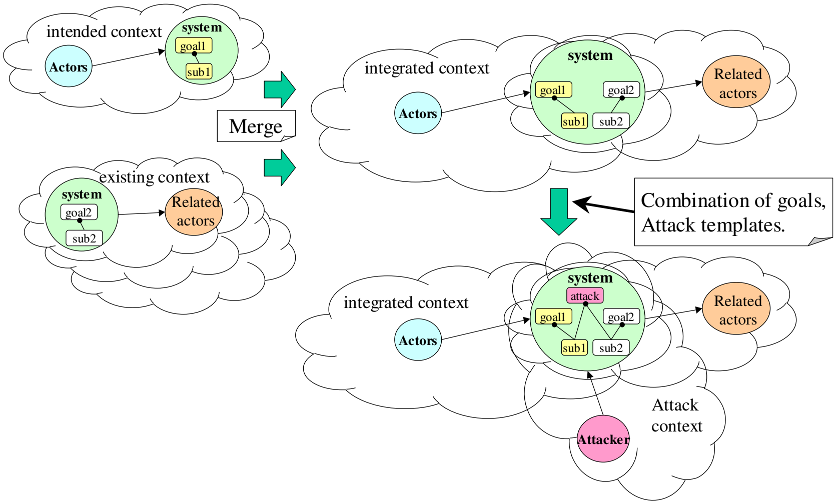


Fig. 1. Overview of the method

3 Method

3.1 Overview

The method contributes to elicit security requirements for a system, which is used in several different activities. We call a model of each activity a context. Figure 1 shows the overview in the method.

1. We temporarily introduce an information system to business or activities in a model. We call such a model “intended context”. We may use patterns we proposed [10] for this introduction.
2. We also develop a model of existing usage of the system in the same way. We call the model “existing context”. The existing context can be developed on the basis of observing the actual activities. Especially in the private usage, insecure applications tend to be used. Incorrect decomposition mentioned below should be embedded in each insecure application.
3. We merge the existing context with the intended context by unifying the system and their related actors respectively. We call the model “integrated context”.
4. We explore suspected attacks on the basis of the combination of achievable goals in the system. To facilitate such exploration, STRIDE patterns [11] are used as templates for concrete attacks.
5. We explore how to mitigate or avoid the attacks. Because goals of both attackers and proper users are represented in goal trees, each goal can be regarded as a logic formula. We try to choose achievable sub-goals so that users’ goals are satisfied and attackers’ goals are not satisfied. We have to reconsider the goal trees or accept the existence of insecure goals when such achievable goals do not exist.

If more than one existing usages already exist, we have to repeat the steps above.

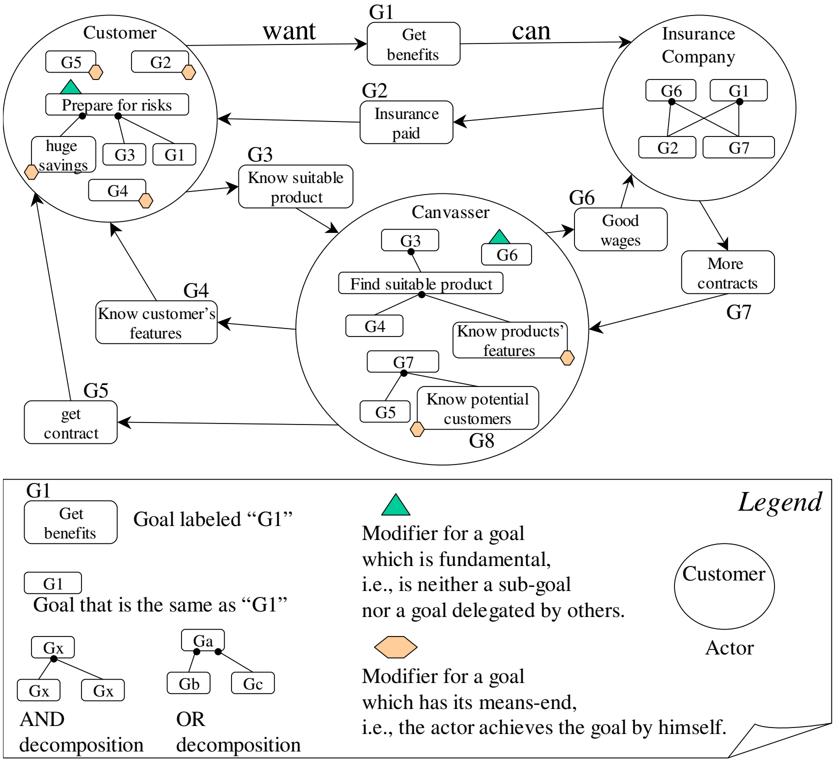


Fig. 2. An example of our simplified i* model about an insurance business. Incorrect goal decompositions are not contained in this example.

3.2 Simplified i* Model

For our security requirements analysis, we use simplified i* notation because we use strategic dependencies and goal hierarchies in each actor. Although most variations of i* contains them, they are not the original idea of the variations but the variations simply adapt them. We simplified original i* as follows. First, we do not distinguish four different types of goals (intentions) such as goals, soft-goals, tasks and resources because there are several constrains to construct goal graphs among such types [12]. For example, a goal cannot be decomposed into other goals directly, but the goal should be refined into other task(s). Second, we do not explicitly specify a task, which is a means for achieving a goal. Instead, we specify an actor has the means to achieve a goal. Third, we specify a goal is fundamental or not. We think a goal in an actor can be categorized into three types: a sub-goal of another goal, a goal delegated by another actor, and the rest. The rest goal can be regarded as a goal, which is initially and fundamentally wanted by the actor. For example, a goal of human "Want to survive" is really fundamental. We think we have to give priority to fundamental goals because the other types' goals are just subcontracts. Fourth, we introduce incorrect goal decompositions to specify a kind of malicious intentions. Although i* is based on the delegation of goal achievement, dependee does not always achieve a delegated goal properly in real life. For example, a meeting participant does not report his vacant schedule correctly if he wants to disturb the meeting initiator.

By using an example in Figure 2, we explain our simplified i* model in detail. An insurance business is modeled in the example. In the model, three actors “Customer”, “Canvasser” and “Insurance Company” are represented in circle. Goals are represented in round-rectangles. In i*, the actor A is called depender and the actor B is called dependee. Such dependency is represented as an actor-goal-actor relationship. For example, an actor “Customer” has a goal G1 “Get benefits”, and another actor “Insurance Company” has a responsibility to achieve the goal. Such dependency is represented in the top of Figure 2. For convenience, we call a relationship between depender and its goal want-relation, and another relationship between the goal and dependee can-relation as shown in the figure. An actor who becomes a dependee of some goals has to achieve the goals. There are two typical ways to achieve them. The first way is to achieve it directly. The second way is to refine and to decompose the goal into several sub-goals, and the actor delegates other actors to achieve each sub-goal. An example of the first way is G2 in the actor “Customer”. The actor “Insurance Company” wants “Customer” to pay insurance every month or year until the contract ends. The goal G2 represents such goal of the “Insurance Company”. The “Customer” will achieve the goal G2 by get a salary for example, but we don’t mention how to achieve G2 in this model. We simply attach a modifier (its shape is hexagon in our model), which clarifies the goal is achieved by the actor. An example of the second way is G1 in the actor “Insurance Company”. The actor “Customer” wants “Insurance Company” to give benefits when he encounters some problem such as a sick or an injury. To achieve the goal G1, the company has to gather money in advance. The company thus has to have enough amounts of contracts (G7) and insurances are paid by each customer according to each contract (G2). Because these goals G2 and G7 cannot be directly achieved by the company, they are delegated to customer and canvasser respectively in this example. We then review the examples of fundamental goals. In our simplified model, a fundamental goal has a modifier which shape is triangle. In the model in figure 2, there are two fundamental goals: “Prepare for risks” of customer and G6 “Good wages” of canvasser. At least in this model, the goal is neither sub-goal of another nor a goal delegated by another actor. The actors fundamentally have such goals because of the reasons outside the model. The fundamental goals are the origin of the goals and their dependencies in the model. Therefore, the model is dramatically changed when some fundamental goals are withdrawn or modified. On the other hand, the other goals and dependencies are variable because there are several different alternatives to achieve the fundamental goals.

In original i*, a goal of a depender is assumed to be properly achieved by a dependee. For example in Figure 2, G3 “know suitable product” of Customer is properly achieved by Canvasser as follows. To achieve G3, the canvasser tries to find suitable product for the customer. He then asks the customer to tell her features such as age and income, and he investigate existing insurance products on the basis of the features. However, a goal of a depender is not always achieved by a dependee when malicious or incapable actors participate in the social dependencies. If a dependee is a malicious human or a system tampered by the human, the dependee intentionally decomposes the goal of a depender incorrectly. Tampering programs executed by users such as cross-site request forgery (CSRF) is typical case.

To represent such incorrect goal-decomposition, we introduce two modifiers to decomposition: one is cut-modifier and another is malicious-modifier. Each modifier is

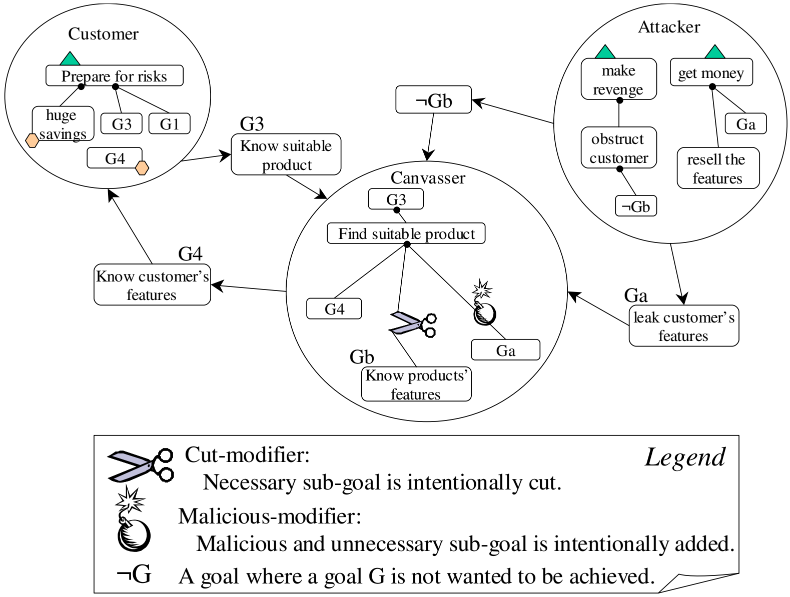


Fig. 3. Another example of our simplified i* model for explaining incorrect goal decompositions

represented in "scissors" and "bomb" icons respectively as shown in Figure 3. We explain how to use the modifiers by using Figure 3. The example in the figure is a modified version of the sub-model in Figure 2. To achieve G3, the canvasser has to achieve both G4 and Gb "Know products' features". However in this example, Gb is not achieved because the decomposition between "Find suitable product" and Gb is cut (by scissors). To realize this situation, inexperienced canvasser is dispatched to this customer. This cutting itself is represented as a goal of "Attacker", and the goal is represented as "¬Gb". The goal "¬Gb" is used as sub-goal to satisfy attacker's goal "make revenge" on the customer. In this example, we assume the attacker personally hate the customer. To achieve G3, the canvasser of course does not have to achieve Ga "leak customer's features". The attacker's threats can terrify the canvasser into achieving Ga when the attacker can take advantage of canvasser's weakness for example. The goal Ga then becomes one of sub-goals of "Find suitable product". This unnecessary goal Ga is used as means for an attack, and the attacker can earn money as shown in this example.

Cut-modifiers are not used so frequently because its super-goals become unachievable and depender can easily know incorrect decomposition in dependee. For example, G3 in Figure 3 becomes unachievable and Customer can know it. However, it is not easy to know sub-goals introduced by malicious-modifiers because its super-goals are still achievable even with the malicious sub-goals.

3.3 Deriving Attacker Context

As mentioned in overview of this method, an actor corresponding to a system contains various goals that do not normally coexist. In addition, systems sometimes contain incorrect goal decomposition where necessary sub-goals are omitted or malicious and unnecessary sub-goals are added. An attacker can thus achieve his malicious goals (attacks) by using such goals. This is an extended idea by Liu et al. [6], [7]. In Liu's idea, a proper actor becomes an attacker, and he uses his achievable goals for his attacks. Our idea is almost the same as the Liu's idea, but attackers' goals can be predicted more than ever because the number of goal-combinations enabling attacks increases.

It is very difficult to predict the malicious intentions of attackers. We think the fundamental goals of attackers will be getting money, harassment, revenge, boasting his power and so on. However, they are too abstract to make the traceability between achievable goals in a system and such attackers' goals. We thus use STRIDE [11] as templates of temporary goals of attackers. STRIDE is a classification of the effects of realizing a threat and is an acronym of the following.

> Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege.

To realize a threat, each effect above requires several means. For example, an attacker can successfully tamper some information when he can interrupt and update it. The attacker can also successfully disclose some information when he can access and send it to somewhere. We find a combination of goals in an actor so that the means of the goals can realize an effect in STRIDE.

3.4 Mitigating or Avoiding Attackers' Goals

To mitigate or to avoid the attacks, we basically disable some of such goals in a system, and find alternatives not to disable goals of proper actors. Even before integrating contexts, actors including a system achieve several different fundamental and delegated goals at the same time. An actor normally hopes such goals in his strategic rationale will be achieved. On the other hand, an actor hopes that goals delegated by attackers, i.e., attacks are unachievable. Because each fundamental and delegated goal in an actor is a goal tree or a goal DAG, it can be represented in a propositional logic formula. The problem for satisfying goals and avoiding attackers' goals can be regarded as satisfiability problem of corresponding formulae. We will introduce the concrete examples of such formulae in the next section.

4 Case Study

In this section, our method is applied to the business for canvassing for the insurance contract to facilitate BYOD (Bring Your Own Device). The idea for using private smart devices in the context of business is called BYOD, and some companies try to facilitate the idea because of its merits related to economizing and usability. The goal of this case study is to illustrate whether our method works well.

Figure 2 shows the as-is model of the business before systems are introduced. In Japan, this type of business is popular in insurance business. An actor "Customer" is a potential customer of the insurance contract. An actor "Canvasser" is a salesperson who explains the insurance products and invites customers to make the contract. Normally in Japan, canvassers are members of specific insurance company. An actor "Insurance company" is the company of insurance products. Because of the fundamental goals "Prepare for risks" of Customer and G6 "Good Wages" of Canvasser, the model in Figure 2 is constructed. Because G6 is delegated to Insurance Company, it decomposes

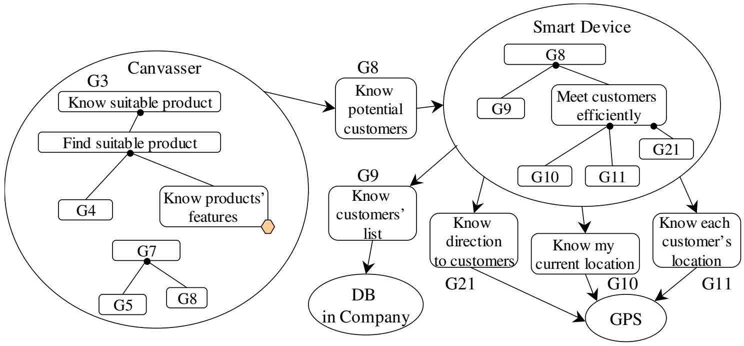


Fig. 4. A part of intended context in section 4.1

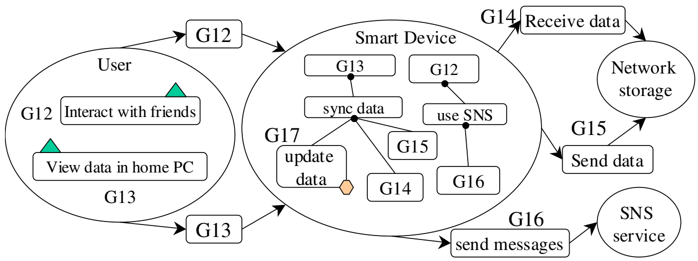


Fig. 5. Existing context in section 4.1

G6 into two sub-goals: G2 "insurance paid" and G7 "More contracts". Each sub-goal is delegated to Customer and Canvasser respectively. G7 is also decomposed into two sub-goals: G5 "get contract" and G8 "Know potential customers" in Canvasser. Because whether G5 is achieved or not depends on Customer, G5 is delegated to Customer. Canvasser achieves G8 by himself and means-end modifier is attached to G8 in Figure 2. For example about achieving G8, he has the knowledge about people in a specific area.

4.1 Supporting Knowledge Acquisition about Potential Customers

Figure 4 shows an intended context after introducing a smart device in Figure 2. Because the device interacts with external servers such as "DB in company" and "GPS" (Global Positioning System), actors corresponding to them are added in the model in Figure 4.

We also ask a canvasser to let us know his private usage of the smart device. In this case, he uses his private device at least for two goals. One is about interacting with his friends via the device, i.e., G12 in Figure 5. Another is about viewing his data in his home PC, i.e., G13 in the figure. Because the former goal is achieved by SNS, an actor "SNS service" and its related functionality are contained in this model as shown in the figure. Because the latter goal is achieved by network storage such as dropbox [1], "Network storage" and its related functionalities are contained in this model as shown in the figure.

[1] `https://www.dropbox.com`

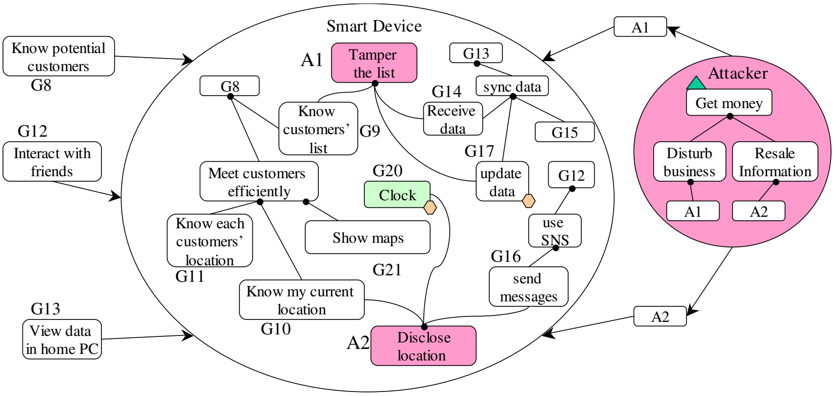


Fig. 6. A part of integrated context (including attack context) in section 4.1

Because "Smart Device" in Figure 4 is the same as one in Figure 5, we developed the integrated context by unifying them. We then explore suspected attacks on the basis of STRIDE mentioned in 3.3. We suspected two attacks as shown in Figure 6. One attack A1 "Tamper the list" can be achieved by G9, G17 and G14 because a program related to network storage can update data in the device and the customers' list is assumed to be stored as data. The contents for tampering the list can be also received by the G14 "Receive data". Another attack A2 "Disclose location" can be achieved by G10, "Clock" and G16 because of the similar reasons for A1. As shown in Figure 6, we suspect the gaps between a fundamental goal of an attacker "Get money" and these concrete attacks, e.g., A1 and A2. We suspect two intermediate goals of attackers: "Disturb business" and "Resale information".

To mitigate or avoid the attacks A1 and A2 in Figure 6, introducing the separation of duty is simple idea. However, required permissions for actual smart devices such as Android phones are too difficult for normal users and we normally have to accept their request for the permission [13]. It is thus hard to introduce clear separation of duty in smart device usage. Another idea for mitigating or avoiding the attacks is to explore alternatives to achieve super goals of goals that are used in the attacks. As mentioned in section 3.4, fundamental and delegated goals can be represented in logic formulas. At the smart device in this case, G8, G12 and G13 are wanted to be achieved, and A1 and A2 are wanted to be unachieved. These goals can be represented in the following formulas.

$$
\begin{aligned}
&G8 \equiv G9 \wedge (G21 \vee (G11 \wedge G10))\\
&G12 \equiv G16\\
&G13 \equiv G14 \wedge G17 \wedge G15\\
&A1 \equiv G9 \wedge G14 \wedge G17\\
&A2 \equiv G10 \wedge G20 \wedge G16
\end{aligned}
$$

To choose an alternative G21 and to disable G10, G8 and G12 can be achieved and A2 can be also unachieved. However, there is no alternative to achieve G8 and G13 and to avoid A1 at the same time in this model. We thus have to find alternatives to G9 and/or G14 respectively for achieving G9 and G13 and for avoiding A1. To explore other attack possibilities, intermediate goals of the Attacker in Figure 6 are useful. For example, tampering the location data can disturb the efficient travel of the canvasser.

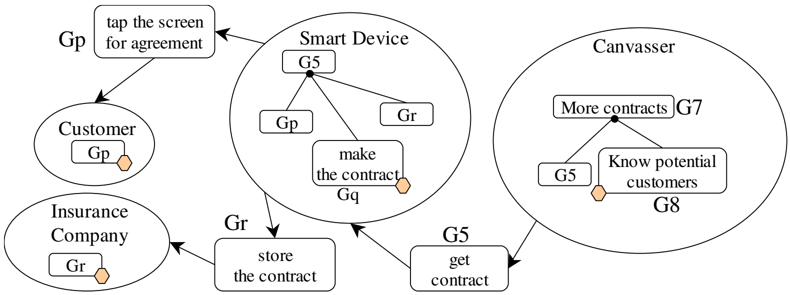


Fig. 7. A part of an intended context in section 4.2

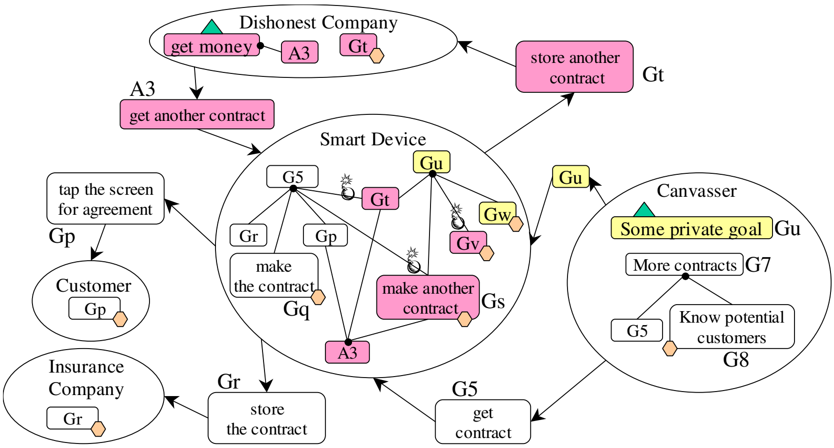


Fig. 8. A part of integrated context in section 4.2

4.2 Easing Reluctance to Make the Contract

Making contract of the insurance is highly reluctant because it traditionally requires a lot of paper works and customers have to sign the documents a lot. We thus try to ease the reluctance by using the smart device. Although the copyright transfer agreement of a technical paper even required written signature, some publishers such as IEEE recently provides the web sites for skipping written signature and we personally become happy without sending the fax to publishers. In the same way, we try to provide insurance customers to make contract easily. In Figure 2, the Customer has to directly satisfy G5 "get contract" by his paper works. To introduce a smart device such as a smart phone, we ease this reluctant works as shown in Figure 7. In the figure, the Customer simply taps the screen of the smart device for his agreement, and a smart device makes the contract document and sends it to the insurance company automatically.

We assume Canvasser satisfies some private goal by some free software on the smart device and the software contains some malicious codes. On the basis of this existing context, we depict the to-be integrated context in Figure 8. In the figure, Gu corresponds to the private usage of the smart device. To satisfy Gu, the device primarily achieves the sub-goal Gw. However, there is a possibility that Gu contains several malicious sub-goals: Gv corresponds to tampering G5, and Gt and Gs correspond to malicious codes embedded in G5. The possibility is predicted on the basis of the permissions given to an application achieving Gu. In the figure, unintended and unnecessary contract (another contract in the figure) is made by the malicious codes. For example, unnecessary goods are bought and paid against the intention of the Customer. As a result, a Dishonest Company, i.e., an attacker can get money by spoofing the Customer, i.e., getting another contract. Goals in the smart device can be represented in the following formulas.

$$\mathrm{G5} \equiv \mathrm{Gr} \wedge \mathrm{Gq} \wedge \mathrm{Gp} \wedge \mathrm{Gt}$$
$$\mathrm{Gu} \equiv \mathrm{Gw} \wedge \mathrm{Gv} \wedge \mathrm{Gs} \wedge \mathrm{Gt}$$
$$\mathrm{A3} \equiv \mathrm{Gp} \wedge \mathrm{Gt} \wedge \mathrm{Gs}$$

Because the goal tree of Gu is predicted on the basis of the permission of its corresponding application, the tree contains malicious goals such as Gt, Gs and Gv. Because of the abilities of such malicious goals, G5 also contains Gt and Gs. The best way to avoid A3 is of course to uninstall the polluted software for achieving Gu, and to install more reliable software that can achieve Gu. However, it is not so easy because an application corresponding to Gu is private one. At least, the boss of the canvasser cannot enforce its uninstall.

4.3 Discussion

For suspecting attackers' goals, STRIDE in section 3.3 seem to works well. To finding the combination of achievable goals for each STRIDE such as tampering, ontological support [14] seems to be effective because the combination for each STRIDE has typical patterns. For example, tampering requires both receiving data from outside and writing data. We want to construct the ontology for such support.

For resolving the security problems, focusing on logic formulas corresponding to goal trees in each actor is simply and effective. However, the problem is not always resolved. For example, an attack A3 in section 4.2 cannot be resolved because of the private reason. One of the simple solutions for this problem is to introduce the separation of duty explicitly. Another solution is to introduce the monitoring mechanism based on our model, and to mitigate the security problems. It is more feasible than introducing rigorous separation of duty because we cannot live in the completely secure environment. We can minimize the activities to be monitored in the existing context based on our model. Each worker can accept such monitoring mechanism because the reasons and the range of monitoring are explained.

We had a workshop to explain our method to security practitioners. Although we did not compare existing i* extensions to our method, at least they understood our method well. From our other experiences [15], we found that practitioners and even students cannot correctly develop full i* model. We thus assume our simplification contributes to motivating practitioners to use our i* variation.

5 Conclusion

In this paper, we present a method for analyzing security requirements for a system, which is already used in several different activities. We use simplified i* model for this analysis. Through a case study of BYOD, we confirmed the method works well. Because the vocabularies in our simplified i* are so limited for security analysis, usual security experts can accept the notation.

In our previous work [16], we presented a method for evaluating a system introduction on the basis of the amount of responsibilities and satisfaction of human in average. Because we defined these metrics on the basis of the structural characteristics of an extended i* model, we can systematically compare an as-is model with its to-be model written in our extended i* model. However, such metrics only focus on positive aspects, and there is no metrics with respect to the security issues. We want to propose and evaluate the metrics for security as well as our current metrics.

References

1. Yu, E.S.K.: Towards modeling and reasoning support for early-phase requirements engineering. In: RE, pp. 226–235 (1997)
2. Yu, E., Giorgini, P., Maiden, N., Mylopoulos, J.: Social Modeling for Requirements Engineering. The MIT Press (2010) ISBN 0262240556
3. Giorgini, P., Massacci, F., Mylopoulos, J., Zannone, N.: Modeling security requirements through ownership, permission and delegation. In: RE, pp. 167–176 (2005)
4. Mouratidis, H., Giorgini, P.: Secure tropos: a security-oriented extension of the tropos methodology. International Journal of Software Engineering and Knowledge Engineering 17(2), 285–309 (2007)
5. Sutcliffe, A.G.: Trust: From cognition to conceptual models and design. In: Martinez, F.H., Pohl, K. (eds.) CAiSE 2006. LNCS, vol. 4001, pp. 3–17. Springer, Heidelberg (2006)
6. Liu, L., Yu, E.S.K., Mylopoulos, J.: Security and privacy requirements analysis within a social setting. In: RE, pp. 151–161 (2003)
7. Liu, L., Yu, E.S.K., Mylopoulos, J.: Secure-i*: Engineering secure software systems through social analysis. Int. J. Software and Informatics 3(1), 89–120 (2009)
8. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., Mylopoulos, J.: Tropos: An agent-oriented software development methodology. Autonomous Agents and Multi-Agent Systems 8(3), 203–236 (2004)
9. Blaine, J.D., Cleland-Huang, J.: Software Quality Requirements: How to Balance Competing Priorities. IEEE Software 25(2), 22–24 (2008)
10. Kaiya, H., Morita, S., Ogata, S., Kaijiri, K., Hayashi, S., Saeki, M.: Model Transformation Patterns for Introducing Suitable Information Systems. In: 19th Asia Pacific Software Engineering Conference, APSEC 2012, Hong Kong, pp. 434–439. IEEE CS (December 2012)
11. Frank Swiderski and Window Snyder. Threat Modeling. Microsoft Press (2004)
12. Yu, E., Castro, J., Perini, A.: Strategic Actors Modeling with i*. In: RE 2008, Tutorial (August 2008)
13. Felt, A.P., Chin, E., Hanna, S., Song, D., Wagner, D.: Android permissions demystified. In: ACM Conference on Computer and Communications Security, pp. 627–638 (2011)
14. Shibaoka, M., Kaiya, H., Saeki, M.: Goore: Goal-oriented and ontology driven requirements elicitation method. In: ER Workshops, pp. 225–234 (2007)
15. Honiden, S., Tahara, Y., Yoshioka, N., Taguchi, K., Washizaki, H.: Top se: Educating superarchitects who can apply software engineering tools to practical development in japan. In: ICSE, pp. 708–718 (2007)
16. Kaiya, H., Morita, S., Kaijiri, K., Hayashi, S., Saeki, M.: Facilitating business improvement by information systems using model transformation and metrics. In: CAiSE Forum, pp. 106–113 (2012)

Towards Engineering Trust-Aware Future Internet Systems*

Francisco Moyano, Carmen Fernandez, and Javier Lopez

Network, Information and Computer Security Lab
University of Malaga, 29071 Malaga, Spain
{moyano,mcgago,jlm}@lcc.uma.es

Abstract. Security must be a primary concern when engineering Future Internet (FI) systems and applications. In order to achieve secure solutions, we need to capture security requirements early in the Software Development Life Cycle (SDLC). Whereas the security community has traditionally focused on providing tools and mechanisms to capture and express hard security requirements (e.g. confidentiality), little attention has been paid to other important requirements such as trust and reputation. We argue that these soft security requirements can leverage security in open, distributed, heterogeneous systems and applications and that they must be included in an early phase as part of the development process. In this paper we propose a UML extension for specifying trust and reputation requirements, and we apply it to an eHealth case study.

Keywords: Trust, Reputation, Requirements Engineering, Secure Design.

1 Introduction

Security is a crucial concern that must be addressed in order to guarantee the successful deployments of FI scenarios [19]. These scenarios usually comprise a huge number of heterogeneous, geographically distributed entities, including human users, which must interact to provide services. The complexity of managing security in these scenarios is aggravated by their dynamic nature, with devices changing, appearing and disappearing along the system lifetime. In these complex and open scenarios, more flexible security solutions, namely soft security mechanisms [14], are required as a complement to the traditional hard security ones: confidentiality, integrity and availability.

Trust is a soft security mechanism that can leverage the security of a system. Even though there is not any accepted definition of trust, it is agreed that it can improve decision-making processes under risk and uncertainty, improving in turn systems security. Reputation, which is a concept strongly related to trust,

* This work has been partially funded by the European Commission through the FP7/2007-2013 project NESSoS (www.nessos-project.eu) under grant agreement number 256980. The first author is funded by the Spanish Ministry of Education through the National F.P.U. Program.

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 490–501, 2013.

can also help in this task. We argue that increasing security in FI applications entails that trust relationships between actors, applications and system environments cannot be taken for granted any more and must be explicitly specified from the very beginning in the Software Development Life Cycle (SDLC). However, security requirements engineering methods often lay trust aside and focus on specifying hard security requirements, such as confidentiality or authorization [7][8]. Even when some social aspects are beginning to be captured at the requirements stages [10][18], the approach towards analysing trust is still naive. This could explain why trust and reputation models have been traditionally added after-the-fac in an ad-hoc fashion, limiting their re-usability and presenting scalability problems [4].

We advocate that a comprehensive analysis of trust and reputation during the initial stages of the SDLC is required. The contribution of this paper is twofold. First, we provide an extension to UML in order to help requirements engineers and software designers to have a clearer understanding of the trust and reputation requirements of the system-to-be; second, we analyse, by means of an eHealth case study, how we can apply this UML profile as well as some considerations when designing trust-aware systems. We choose UML because it is a *de facto* standard in the industry and because other relevant security-oriented profiles exist that could be potentially integrated with ours.

The rest of the paper is structured as follows. Section 2 reviews some related work. In Section 3 we provide a domain analysis in the field of trust and reputation. The extensions performed on UML are explained in Section 4, whereas Section 5 applies these extensions to an eHealth scenario. Finally, Section 6 presents the conclusion and some lines for future research.

2 Related Work

There are several works that consider security requirements at the early stages of the SDLC. Some of these works focus on detecting possible attacks on the system [16][15]. In others, the emphasis is on modeling security requirements, such as confidentiality or authorization. This is the case of UMLsec [7] and SecureUML [8], two UML profiles that include security constraints and annotations into the diagrams. Other works aim to integrate the notion of risk into the requirement analysis stage [9] in order to assess whether the risk level of some unwanted incidents is beyond an acceptable threshold.

The contributions mentioned up to now focus on hard security requirements or risk, but they usually lay trust aside. In addition to traditional policy languages for distributed trust management [2][6], there are other works that focus on trust in early stages of the SDLC. Mouratidis and Giorgini [10] present Secure Tropos, a methodology that extends the Tropos methodology in order to enable the design of secure systems. Actors in Tropos may depend on other actors in order to achieve a goal, and these social relationships are captured by the methodology. In a similar direction, Lamsweerde and Letier present KAOS [18], a comprehensive goal-oriented methodology to elicit the requirements of a socio-technical system. All these contributions put forward the idea of capturing social aspects, but the notion of trust and its influence on the information systems is barely explored. This is partially covered by Pavlidis, Mouratidis and Islam [12], who include trust-related concepts in Secure Tropos.

The work by Chakraborty and Ray [3] bridges a gap between traditional security requirements modeling and soft-security considerations by incorporating the notion of trust levels into the traditional Role-Based Access Control model. These levels are measured by means of a trust vector, where each component in the vector is a factor that influences trust, such as knowledge or experience.

In general, the aforementioned works usually fail in capturing and making explicit all the trust relationships, and above all, how trust and reputation can be used by the system-to-be. The closest contribution to our work is the one by Uddin and Zulkernine [17], who present a UML profile for trust called UMLtrust. They provide extensions, as we do, to some UML diagrams in order to represent trust information. Their approach and focus is, however, different than ours. First, their primary concern is reasoning about *trust scenarios*, without making explicit which are the trust relationships in the system. Also, they do not address reputation, whereas it is a primary concern for us. We also provide more details on how trust and reputation can be computed and the factors (e.g. variables and attributes) that will be taken into account for this computation. We also show how trust can influence at the infrastructure level by means of deployment diagrams. However, our trust analysis is in general at a higher level of abstraction, without delving into the details of class attributes and methods, which is something that UMLtrust requires. As a conclusion, we think that both works are complementary and can help each other in providing a more comprehensive vision of trust in the system for designers and developers.

3 Trust and Reputation

There are many definitions of trust, and one often cited is the one by Gambetta [5]: *'trust is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action[...]'*. Reputation is defined by the Concise Oxford Dictionary as 'what is generally said or believed about a person or the character or standing of a thing', being a more objective concept that trust.

A trust conceptual model, adapted from the one in [11], is shown in Figure 1. Many of these concepts are included in the UML profile presented in the next section. Just to mention some details, it is important to differentiate between a trust factor, a variable and an attribute. A factor is an application-independent attribute (e.g. the trustor's disposition to believe in others' capability). An attribute captures the notion of factor and extends it with application-specific information too (e.g. how much a customer has paid for a book). At the software level, one or more attributes are represented by variables, which forms the core of a trust metric. The dimension of a trust or reputation value refers to whether the final score is just one number (or label), or a tuple of numbers or labels. Finally, for simplicity sake, some arrows are not depicted. For example, there would be an arrow from *Reputation* to *Value* and from *Variable* to *Value*.

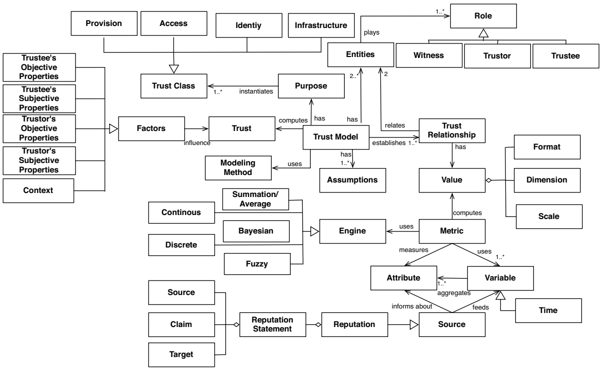


Fig. 1. Trust Conceptual Model (summarized)

4 UML Profile for Trust and Reputation

This section presents an extension of UML that aims to ease the specification and initial design of trust-aware systems and applications. We consider indispensable the use of some behavioural diagram, such as activity diagrams, even though we do not propose extensions for them. This diagram should represent the interaction patterns between the trust and business logic of the application, and should make clear which actor initiates a trust event, how this event is triggered and what their consequences are[1].

Each extended diagram is further explained in the following sections.

4.1 Use Case Diagram

The goal of use case diagrams in the context of trust is to depict, at a glimpse, the trust relationships that exist between the different actors in the system. There is however more interesting information that we can represent in this diagram. For example, we could make explicit which actors can make a claim about which other actors, thus incorporating reputation information in the diagram. The extensions performed on the use case diagram are summarized in Table 1.

Trustor, *trustee*, *witness*, *source* and *target* are roles that actors can play in the system. Trust relationships are made explicit by means of the extension *trusts*, whereas *claims* represent that a given *source* can make a claim about a given *target*. As the ultimate goal of trust is aiding in making a decision, we also add the *decides* connector, which captures the idea that a use case can be affected by trust or reputation information. An actor could perform the same use case in different ways (or even could decide not to perform it at all), and this decision can be influenced by trust or reputation information.

[1] A trust event is any occurrence in the system that triggers a reputation or trust relationship update.

Table 1. Use Case Diagram Extensions

Stereotype	Base Class	Explanation
Trustor	Actor	Actor playing the trustor role
Trustee	Actor	Actor playing the trustee role
Witness	Actor	Actor playing the witness role
Source	Actor	Actor capable of making a claim
Target	Actor	Actor capable of receiving a claim
Trusts	Connector	Trust relationship
Claims	Connector	Source makes a claim about a Target
Decides	Connector	Use case affected by a trust/reputation decision

In addition to the previous UML extensions, we define two adornments: *decision criteria* and *context*. The former is used to annotate the *decides* relationship between an actor and a use case, and it specifies whether the decision in that use case is based on trust or reputation. The latter annotates *trusts* and *claims* relationships and specifies their context. This captures the idea that trust and reputation are context-dependent.

4.2 Class Diagram

Class diagrams can provide more insight in certain aspects of trust and reputation. The stereotypes used to extend class diagrams are depicted in Table 2. We find the same stereotypes that in the use case diagram extension regarding the roles of the actors. Also, we find *TrustRelationship*, which represent the trust relationship between a *trustor* and a *trustee*, and *Claim*, which captures the notion of a claim made by a *source* entity about a *target* entity. We add also three important notions for the evaluation of trust and reputation, namely *TrustEngine*, *ReputationEngine* and *Variable*. They represent how trust and reputation are computed, and the variables considered for such computation.

Tagged values are used in order to define more precisely the aforementioned concepts. The list of tagged values is shown in Table 3. Just to mention some of them, *subjective properties* refer to a list of beliefs of the trustor regarding a trustee, whereas *objective properties* represent a list of trustee's properties that can be (more) objectively measured (e.g. reliability or certification by a Trusted Third Party). *Dimension* is the number of components of a trust or reputation value, and *how* specifies whether the value of a variable is explicitly assigned (interactively) by the actor or is monitored by another system.

Note that some of these tagged values could be almost directly mapped to attributes of design classes, whether others are just informative and require further refinement at design stage. For example, *attribute* represents the attribute(s) captured by a variable. This information might be useful for aiding designers to keep in mind what the variable actually should represent, but could hardly be directly mapped to a class attribute.

Table 2. Class Diagram Extensions

Stereotype	Base Class	Explanation
Trustor	Class	Actor playing the trustor role
Trustee	Class	Actor playing the trustee role
Witness	Class	Actor playing the witness role
Source	Class	Actor capable of making a claim
Target	Class	Actor capable of receiving a claim
TrustRelationship	Class	Trust relationship between trustor and trustee
Claim	Class	Claim that a source makes about a target
TrustEngine	Class	Engine in charge of updating a trust relationship
ReputationEngine	Class	Engine in charge of computing a target's reputation
Variable	Class	Variable used by a trust or reputation engine

Table 3. Tagged Values for Class Diagrams

Value	Class	Explanation
type	Trustor, Trustee, Witness, Source, Target	The type of actor (i.e. human, system)
subProp	Trustor	Subjective properties
objProp	Trustee	Objective properties
context	TrustRelationship, Claim	Context
dimension	TrustRelationship, Claim	Dimension of a trust relationship or a claim
scale	TrustRelationship, Claim, Variable	Upper and lower bounds
default	TrustRelationship	Default value
format	TrustRelationship, Claim	Quantitative vs. qualitative
display	ReputationEngine	Visualization by user actors
engine	Engine	Type of computation engine
variables	Engine	List of variables used by the engine
attribute	Variable	Attribute(s) captured by the variable
source	Variable	System or actor that triggers the variable update
how	Variable	Assigned vs. monitored

4.3 Deployment Diagram

Deployment diagrams are useful as they represent the software from the infrastructure point of view, and they show valuable information in terms of trust and reputation. Very often, trust and reputation must be considered not only at the application level (trust among actors or among software components), but also at the infrastructure level [13]. Platforms and networks can trust each other and they can even hold reputation values. This is particularly useful when designing large-scale distributed systems, where a given processing node (e.g. a mobile phone or a server) can choose among different nodes in order to collaborate or communicate information.

The extensions performed on deployment diagrams are shown in Table 4. We can specify which node acts as reputation manager in a centralized reputation model. Reputation managers compute reputation, store it, and distribute it (or just publish it) when necessary. The *decides* stereotype captures the decision process made by one entity (processing node) when communicating with other processing nodes. As in the case of use case diagrams, this stereotype can be adorned in order to make explicit whether this decision is based on trust or reputation with *decision criteria*. Finally, we also add a tagged value *entities* to specify the reputation of which entities the reputation manager will store.

Table 4. Deployment Diagram Extensions

Stereotype	Base Class	Explanation
ReputationManager	Node	Node that acts as reputation manager
decides	Connector	Trust-based decision

The next section puts all the concepts discussed in this section together by applying them to an eHealth scenario.

5 Case Study

In order to consider trust and reputation requirements early in the SDLC, we will present in this section how we can apply the information provided in Section 4 to a real scenario. The case study comes from the NESSoS project[2] and belongs to an eHealth scenario as described in a project deliverable [1].

The case study presents a patient monitoring scenario, which aims to collect health-related data independently of the location of the patient. This is useful for patients, who can receive immediate feedback under critical situations and be assisted by physicians at any moment and place. In order to make this scenario feasible, the patient must wear a device capable of measuring vital signs (e.g. blood pressure). This device must be able to send this information to other systems that will show it to physicians for monitoring purposes.

The goal is to build a web application through which the physician and the patient can interact in a trusted way. In this application, the physician can add and remove a wearable device to the system, start the process to assign the device to a patient, configure both critical and uncritical alerts, ask patient consent to use his data for research purposes, create an advice for the patient based on the patient's data, demand an immediate reading from the wearable and start the process to change a patient's wearable. Patients can configure uncritical alerts, ask for second opinions (to other physicians), accept or deny consent, show the physician's advices, complete the device assignment process started by the physician and demand a physician change.

Even though there are important hard security requirements, the application must also be trusted, in the sense that physicians and patients must be confident that the application is performing well and that they can trust the information provided by other entities. We propose using the aforementioned UML profile in order to consider trust and reputation requirements early in the SDLC.

A possible trust-aware use case diagram is shown in Figure 2. We state that there is a trust relationship between the patient and the physician. The patient plays a *trustor* role and the physician plays a *trustee* role. In addition, there is a *trusts* connector, which is adorned by the *context* where this trust relationship is set, namely *monitoring*. There is another trust relationship between the physician (who therefore also plays a *trustor* role) and the wearable. The patient also plays the *source* role and can therefore make claims (*claims* connector) about the physician, who plays in this case the *target* role.

[2] `http://www.nessos-project.eu`

Up to now, we have defined the main actors, the trust roles they can play, and the trust relationships and possible claims that the application considers. We also need to include for which purpose this information is going to be used, and this is the role of the *decides* connector. Just to mention two examples, the patient may decide to ask another physician for second opinion. In order to decide who this other physician is, he uses reputation information about the physician (annotation *decision criteria*). Also, the physician may ask for a new wearable if his trust in the actual wearable falls below a certain threshold. Thus, we are using trust and reputation to help actors to make decisions at runtime.

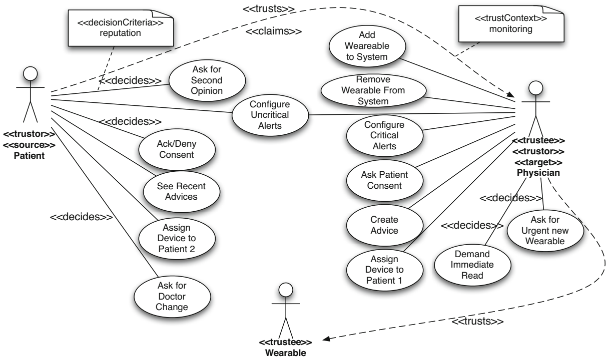


Fig. 2. Trust-aware Use Case Diagram

Claims and trust relationships can be further refined in trust-aware class diagrams, as shown in Figure 3 and Figure 4[3]. Regarding the patient monitoring relationship, we specify the context of this relationship (which should be consistent with the context in the use case diagram), the dimension and format, which are one and numeric in this case, the scale, which is the interval $[0, 1]$, and the default value, which is 0.5. Thus, every trust relationship between a patient and a physician will be assigned by default (i.e. at bootstrap phase) the value 0.5 and will take values between 0 and 1 along the application life. Also, we specify some information regarding the trustor and the trustee. In this relationship, the trustor is a human actor and has a subjective property that influences on the trust relationship: *capability belief*. This means that the belief that the patient has in the capability of the physician must be considered when setting the trust relationship, as stated also by the *trust engine* that updates the trust relationship. This engine uses a continuous engine (meaning that it will yield a continuous value by aggregating continuous variables). The list of variables used by the engine are the reputation of the trustee, the belief of the trustor, and the trustor's quality feedback, which is represented by the claim in Figure 4. Note that in the case of a claim, the *reputation engine* computes reputation for a given target, and not for a trust relationship. The reputation engine gathers

[3] We do not depict the trust relationship physician-wearable due to space limitations.

the claims that different patients make about a given physician and computes a final reputation using an average. In addition to the claims, *time* is also used to derive this reputation value, which will be displayed by a *3 stars* notation.

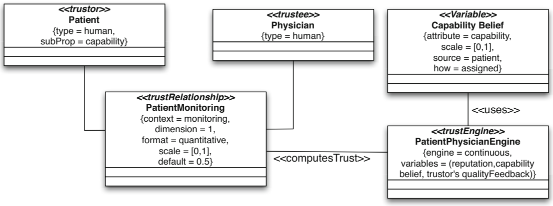


Fig. 3. Patient-Physician relationship

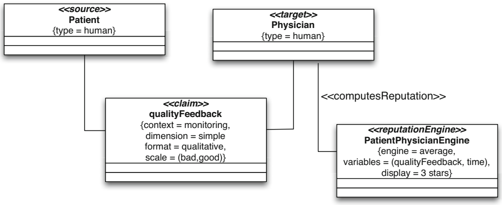


Fig. 4. Quality Feedback Claim

For every variable defined in the engine, we can define a new *Variable* stereotype and specify some of the important properties of them. Figure 3 shows that capability belief, assigned by the patient, will take a value in the interval $[0, 1]$ and should capture the attribute *trustor's capability belief*. In the *physician-wearable* relationship[3], the physician triggers a system that monitors the wearable reliability variable, where the attributes captured by this variable are reliability and accuracy of the provided data.

Note that from the class diagrams information, especially after identifying the variables that we need, we can go back to the use case diagram during the second iteration and add new use cases that should be included. We do not depict them due to space limitations but basically, the patient should have means of rating a physician and to set the physician preferences. This last use case captures the capability belief, as the preference list will likely be made by the patient in terms of this capability belief about the physicians. Finally, the physician should be able to measure the wearable reliability.

How the business and trust layers of the application interact may be a valuable information for designers. This can be depicted by a behavioural diagram, such as an activity diagram. The goal is to represent which actor can trigger a trust event and how, and what are the consequences of that trust event. We propose using swim lanes in order to make clearer the responsibilities of actors, the business logic and the trust logic in the whole application. Figure 5 depicts the trust event triggered when the patient asks for a second opinion.

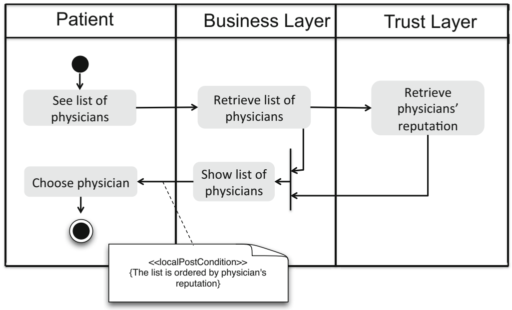


Fig. 5. Activity Diagram for Use Case *Ask for Second Opinion*

The basic deployment for this application, without considering trust information, consists of a sensor that communicates with a wearable, which in turn, aggregates the information and sends it to a front-end server running the application. This front-end server will send the information to a back-end server that will store it into the patient's Electronic Health Record (EHR) and that executes a configuration application (i.e. to configure certain aspects of the application) only available to administrators. Figure 6 shows a trust-aware deployment diagram. The wearable device can decide, based on the front-end server reputation, to which server to send information. The same happens between the front-end server and the back-end server. Of course we are assuming that the final deployment will consist of, at least, two front-end servers and two back-end servers. Otherwise, a decision is not possible. We can also make explicit on which node the reputations for different entities in the system will be stored (i.e. assuming a centralized reputation model). In this case, a node is reserved to play the role of a reputation server that will store the reputation values for physicians, the front-end servers and back-end servers.

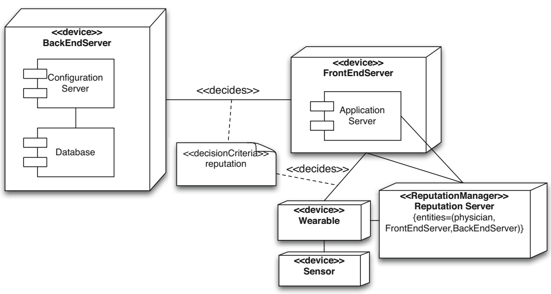


Fig. 6. Trust-aware Deployment Diagram

6 Conclusion

Trust and reputation can be powerful mechanisms to improve security in complex, distributed systems and applications. We have proposed an extension to UML and some design guidelines to help requirements engineers and software designers to have a clearer view on trust requirements. This is not a straightforward task, as the concept of trust itself is difficult to grasp, and as there is an important gap between the social notion of trust and its software representation.

Our goal with this paper has been to continue bridging this gap, even though much work still remains to be done. First, the profile should be further extended in order to represent policies, credentials and trusted third parties, which constitute the roots of many trust management systems nowadays. The profile should also allow representing how trust information can be propagated between actors in the system. Trust derivation from lower software abstractions (e.g. trust among components) to higher level abstractions (e.g. trust among processing nodes), if possible at all, is an interesting field that requires much further exploration. Finally, there is a need for defining the semantics and constraints of each syntactic element. Tool support is then required to check compliance with these constraints and to derive design patterns and code from the specification. In this direction, how to integrate our approach with existing frameworks (e.g. UMLsec) should be analysed.

References

1. Initial version of two case studies, evaluating methodologies. NESSoS Deliverable 11.3 (October 2012)
2. Blaze, M., Feigenbaum, J., Lacy, J.: Decentralized Trust Management. In: Proceedings of the 1996 IEEE Symposium on Security and Privacy, SP 1996, pp. 164–173. IEEE Computer Society, Washington, DC (1996)
3. Chakraborty, S., Ray, I.: Trustbac: integrating trust relationships into the rbac model for access control in open systems. In: Proceedings of the Eleventh ACM Symposium on Access Control Models and Technologies, SACMAT 2006, pp. 49–58. ACM, New York (2006)
4. Farmer, R., Glass, B.: Building Web Reputation Systems, 1st edn. Yahoo! Press, USA (2010)
5. Gambetta, D.: Can We Trust Trust? In: Trust: Making and Breaking Cooperative Relations, pp. 213–237. Basil Blackwell (1988)
6. Grandison, T.: Trust management for internet applications. PhD thesis, University of London (July 2002)
7. Jürjens, J.: UMLsec: Extending UML for Secure Systems Development. In: Jézéquel, J.-M., Hussmann, H., Cook, S. (eds.) UML 2002. LNCS, vol. 2460, pp. 412–425. Springer, Heidelberg (2002)
8. Lodderstedt, T., Basin, D., Doser, J.: SecureUML: A UML-Based Modeling Language for Model-Driven Security. In: Jézéquel, J.-M., Hussmann, H., Cook, S. (eds.) UML 2002. LNCS, vol. 2460, pp. 426–441. Springer, Heidelberg (2002)
9. Lund, M.S., Solhaug, B., Stølen, K.: Model-Driven Risk Analysis - The CORAS Approach. Springer, Heidelberg (2011)

10. Mouratidis, H., Giorgini, P.: Secure Tropos: a Security-Oriented Extension of the Tropos Methodology. International Journal of Software Engineering and Knowledge Engineering 17(2), 285–309 (2007)
11. Moyano, F., Fernandez-Gago, C., Lopez, J.: A Conceptual Framework for Trust Models. In: Fischer-Hübner, S., Katsikas, S., Quirchmayr, G. (eds.) TrustBus 2012. LNCS, vol. 7449, pp. 93–104. Springer, Heidelberg (2012)
12. Pavlidis, M., Mouratidis, H., Islam, S.: Modelling Security Using Trust Based Concepts. IJSSE 3(2), 36–53 (2012)
13. Ramchurn, S.D., Huynh, D., Jennings, N.R.: Trust in multi-agent systems. Knowl. Eng. Rev. 19(1), 1–25 (2004)
14. Rasmusson, L., Jansson, S.: Simulated social control for secure internet commerce. In: Proceedings of the 1996 Workshop on New Security Paradigms, NSPW 1996, pp. 18–25. ACM, New York (1996)
15. Schneier, B.: Attack Trees: Modeling Security Threats. Dr. Dobb's Journal (1999)
16. Sindre, G., Opdahl, A.L.: Eliciting security requirements with misuse cases. Requir. Eng. 10(1), 34–44 (2005)
17. Uddin, M.G., Zulkernine, M.: Umltrust: towards developing trust-aware software. In: Proceedings of the 2008 ACM Symposium on Applied Computing, SAC 2008, pp. 831–836. ACM, New York (2008)
18. van Lamsweerde, A., Letier, E.: Handling Obstacles in Goal-Oriented Requirements Engineering. IEEE Trans. Softw. Eng. 26(10), 978–1005 (2000)
19. van Rooy, D., Bus, J.: Trust and privacy in the future internet - a research perspective. Identity in the Information Society 3(2), 397–404 (2010)

Security Model for Large Scale Content Distribution Applied to Federated Virtual Environments

Adam Wójtowicz

Department of Information Technology, Poznań University of Economics
ul. Mansfelda 4, 60-854 Poznań, Poland
awojtow@kti.ue.poznan.pl

Abstract. In federated multimedia systems new services can be dynamically added or updated, thus a synergy effect related to integration of distributed communities of users and service providers can be observed. However, the inherent security limitation of such systems is implied by malicious host problem, particularly the risk that host software would be modified in order to e.g. violate data confidentiality. In the proposed model the distributed content consumers are provided with encryption scheme securing the confidentiality and integrity of the content roaming with them from host to host e.g. in federated virtual environment. The decryption keys, shared with threshold schemes, are produced in particles that correspond to the subsets of the multimedia content with respect to its structure. The scenes can be reconstructed collectively, but in a selective manner, according to the user privileges. In consequence, the model allows for placing content safely on virtual environment hosts and mitigates the problem of the host code that can be malicious.

Keywords: secure content distribution, secret sharing model, federated system security, multimedia systems.

1 Introduction

Interactive scenes are attractive content form for the modern digital citizens [1], e.g. scenarios based on 3D models of virtual museum objects [2]. Promising container technology for such content is federated virtual environments [3][4] such as Hypergrid [5], for two main reasons. First, they are based on open architecture – new regions on new hosts can be dynamically added to the grid by third parties. Such openness provides a developing market – growing population of active users. Second, they are based on functional, standardized, open, efficient and constantly developing software engines like OpenSim [6], and corresponding client browsers [7].

However, there is a significant obstacle for moving scenes of virtual museums or other content providers to the grid of federated regions (hosts). It is the problem of the data security [8][9], particularly the threat of piracy. In order to participate and contribute to virtual environments, publishers, content creators as well as users need to be sure that their rights to the content will be preserved. It requires assuring confidentiality and integrity of the scenarios that are distributed over the grid, which is hard,

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 502–511, 2013.

mainly because of the malicious host problem [10]. It cannot be assumed that host software has not been modified in order to make illegal copies of digital items that constitute users inventory [11], e.g. for the purposes of illegal pirate distribution. In case of the Hypergrid this is caused by the fact that federated architecture [12] requires user inventory (digital items) to be accessible for target hosts as the user roams from host to host, in order to be properly simulated, rendered or used. The simple authorization mechanism for Hypergrid described in [13] does not solve the problem, since obviously it is assumed that the host visited by the users is an “authorized consumer of the resources”. The malicious host problem has been described and it is partially solved only if specific conditions are fulfilled or for specific sub-problems [14].

From the content publisher perspective, the possible partial workaround of malicious host problem in the grid is to set up an own region serving 3D scenes on the own host. However, such theoretically natural solution has a number of practical drawbacks (e.g. for a small or medium virtual museum), which include: uncertainty whether the large enough number of users will be willing to visit the region; lack of third party content that is related to the owner content, constitutes its natural context, and can additionally attract visitors; the time and cost effort related to the region promotion; the time and cost effort related to the region hosting and maintenance. Moreover, using institution-hosted regions protects only the institution’s rights to the content. Users rights to the content are still not protected, since they cannot be sure whether confidentiality of their roaming inventories is preserved on remote hosts, unless they fully trust the hosting institutions.

Taking all these factors into account, in this paper the solution to the described problem is proposed. The main idea of the proposed model is providing the distributed community of users with the cryptographic means transparently integrated with virtual environments software, allowing to selectively reconstruct the safely distributed scene subsets. The model enables placing 3D scenes in existing popular regions that attract a large number of visitors and provide rich context for the distributed content, and at the same time mitigates the problem of the host code that controls the region and can be potentially malicious. The software solution design is proposed along with the usage scheme that is based on it.

2 Related Work

Recently, a significant development of the federated virtual environments based on open architectures can be noticed, particularly in open source software communities. Good representatives are platforms based on OpenSimulator [6] engine implementation, such as OSgrid/Hypergrid [15][5], where new regions on new hosts can be dynamically added to the grid. A roaming data processing model and external openness makes federated virtual environments inherently insecure which affects trading of digital goods and their usage control. Their access control mechanism can be bypassed, e.g., by using “copybot” software simulating legitimate applications that perform uncontrolled operations to copy user assets while a user visits a virtual environment region (host) that is running a malicious code [11].

Open Cobalt [16], another open source platform for constructing, accessing, and sharing virtual environments, makes it possible to hyperlink virtual environments using 3D portals to form a large distributed network of interconnected collaboration spaces. It does not require centralized servers and the processing is distributed in a P2P manner. From the security analysis point of view, interesting element of such approach is reduction of reliance on error-prone server infrastructures by using a peer-based messaging protocol. However, here the problem of untrusted clients appears, which is hard to solve, as the problem of malicious hosts.

One of the other mature open source platforms is Open Wonderland [17] supporting creation of interactive and dynamic content. For access control, any digital object can be associated with an access control list (ACL) to control which users can view or manipulate or edit the object. Here ACLs are hierarchical, so access can be applied to a single object in a space or to all objects within enclosing 3D structure. However, contrary to Open Simulator, Open Wonderland is not based on the paradigm of federalization and distribution, thus it suffers all the limitations of the centralized system.

Solutions grown from the programming platforms for 3D content should be mentioned as well, i.e. X3D-based collaboration servers, such as BS Collaborate [18]. However, they provide only limited security measures and do not enable creating federated environments, but represent traditional, client-server approach.

The techniques that potentially can be used to make grid data confidential, and, at the same time, allowing computation on, or partial access to the data are, respectively, functional encryption [19] (generalization of predicate encryption [20], attribute-based encryption [21], and identity-based encryption [22]) and structured encryption [23][24] (generalization of searchable encryption [25]). These techniques are useful, e.g. in the cloud computing data processing schemes [24], where data remain encrypted on the host after querying or calculating. However, they cannot be applied directly to the multimedia data in federated virtual environment. This is caused by the fact that the dynamically added untrusted hosts are required to have full access to the scenes and user inventories in order to simulate physics and behavior while the users are interacting with each other or are roaming from host to host.

From the wide spectrum of techniques for providing data confidentiality, threshold secret sharing has been chosen in the proposed approach [26][27][28]. Originally it has been designed for the applications where a number of parties, that do not fully trust each other, have to collectively make decisions or gain access to a secret. Since in many applications unanimous decisions or even full presence cannot be assumed, threshold secret sharing schemes allows at least k of n parties to reconstruct secret message. In organizations secret information can be shared between different groups of users (on homogenic layers, e.g. strategic, tactical) in the same way or differently on each layer [29]. In different layers the same or different information can be shared. Reflecting user hierarchies in the secret sharing consists not only in simple providing the users from the higher hierarchy levels with higher number of shares, but also in producing the shares of the different quality for the levels. Moreover, methods allowing the users from higher levels of the hierarchy to delegate the rights to the secrets have been developed. According to the author's best knowledge the threshold secret sharing has not been used to secure the distribution of the multimedia content in the virtual environments, where it cannot be assumed that users keep acquired data confidential or, even worse, the problem of malicious host appears.

3 Model for Mass Scale Content Distribution

3.1 The Idea

The main idea of the proposed solution is providing the distributed communities of content creators and consumers with the cryptographic means transparently integrated with federated system, allowing for the safely distributed scene subsets to be selectively reconstructed (**Fig. 1**). As an initial step all the elements of the scene and their relations are symmetrically encrypted by the publisher, and the ability to decrypt it is dependent on the submission of the secret keys. The secret keys are split in advance using the threshold secret sharing algorithm for a number of shares. Each shareholder obtains a number of shares proportional to the class of the acquired ticket and corresponding to his or her usage rights to the subsets of the content. In order to start the interactive exhibition, the k of n, or more, shareholders are needed. The keys are produced in particles that correspond to the subsets of the original scene with respect to its structure, in order to provide the ability to reconstruct the scene in a selective manner. The k number of tickets is a parameter of the algorithm (and can be calculated e.g. to create a "critical mass" for which the event is justified for economical reasons). After the shareholders with eligible key shares gather at freely chosen host and decide to start an event, reconstruction of keys is deployed, the scene is decrypted and the exhibition can be run.

Simple scene encryption would not solve the problem since a single ticket holder with the decryption key could illegally distribute the scene. Access control mechanisms designed for the interactive and distributed content, such as proposed in [30][31][32][33][34], also cannot be applied, since they require predefined trusted host, which does not match open grid specificity.

3.2 The Application in Federated Virtual Environments

The proposed concept of the software framework is composed of four main components: a scheme for threshold scenes sharing, layer of integration with virtual environment infrastructure, publisher tools, and a scheme for updating shares.

The first component and the core of the proposed solution is a new scheme for threshold secret sharing adapted to structured and interactive 3D scenes. It is based on existing threshold algorithms, but modified to support specific data structures of the 3D scenes (geometrical models, behaviors/scripts, composability structures). It employs cryptographic keys distribution scheme. The implementation is deployed on the side of virtual museum software (encryption, key splitting), as well as on the side of the region host software (key reconstruction, decryption).

The second component of the solution is a software layer that integrates reconstructing/decrypting algorithms described above with Hypergrid software, or potentially other open virtual environment platform. The integration is transparent (automatic decryption and secret reconstructing on-the-fly) from end-user point of view.

The third component of the solution is a set of publisher tools that allow for selective encryption and sharing of the scene. It integrates splitting/encrypting algorithms with GUI software that enables interactive structuring of the 3D scene. The integration is transparent (automatic encrypting and secret splitting on-the-fly) from the end-user point of view.

The fourth required component is the software implementing scheme of updating shares. It is needed for handling long-term users migration, and is based on proactive threshold secret sharing. The implementation is deployed on the side of the OpenSim-based host software.

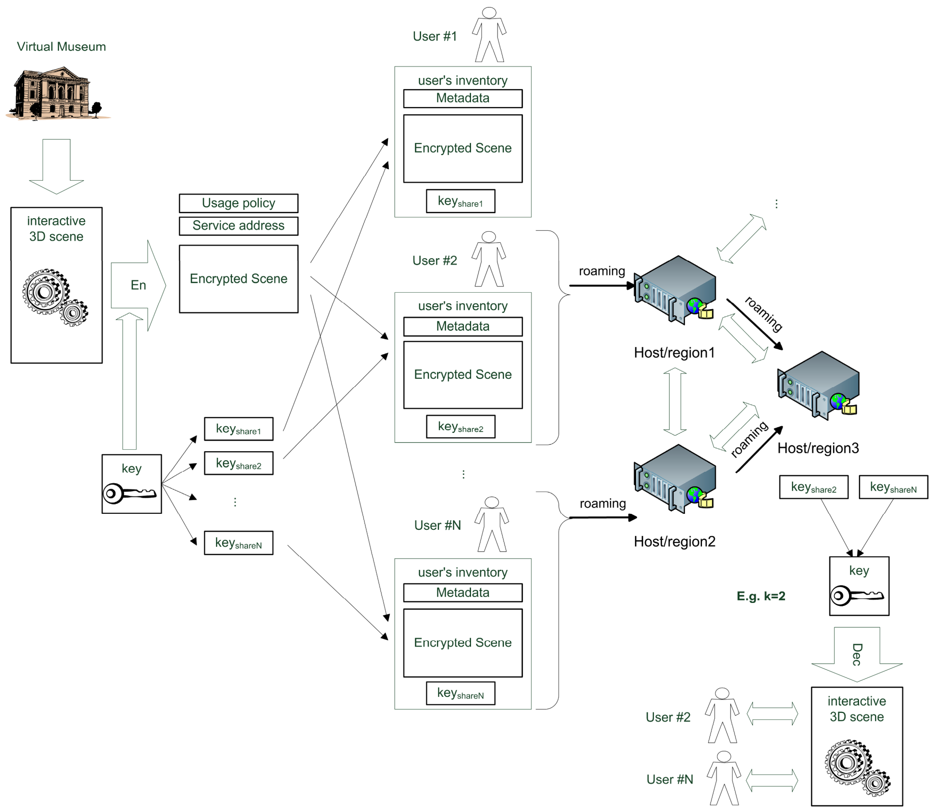


Fig. 1. Scheme for scenes sharing in Hypergrid. Simple example for k=2.

3.3 Structured Content Sharing

In the simplistic example depicted in the **Fig. 1** the whole scene has been shared as a single digital item. However, in the real applications only specific subset of the scenes is shared among different groups of users (c.f. **Fig. 2**), according to users privileges (or roles when RBAC is applied). These subsets can be composed of elements, i.e. scenes, sub-scenes, geometrical models, scene scripts, behavioral scripts, as well as relations between these elements.

In the example in **Fig. 2** two users have shares in two different subsets of the scene. Since scene subset #2 is the superset of scene subset #1, if k=2, the User #1 and User #2 have right to reconstruct together the decryption keys and decrypt the scene. However, their right is limited only to the range of Scene subset #1. Thus, there is a need for applying cryptographic mechanism for selective scene sharing with respect to scene structure. Therefore, the keys are produced in particles that correspond to the elements of the original scene and to relations between them (there are separate keys for relations). Token-based structured encryption and searchable encryption techniques, having applications in the cloud computing [24] which however differ from the open federated grid, are conceptual inspiration here.

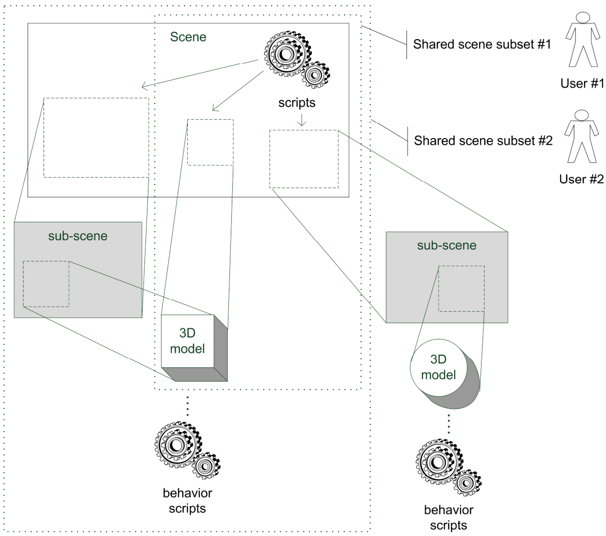


Fig. 2. An example of scene subsets to be shared in case when one scene is the subset of the other

An additional functionality intended to promote the content usage is introduced taking advantage of the features of threshold secret sharing. It assumes that shares are based on variable *k* value for fixed secret, which means that the lowest *k* value is needed to reconstruct scene skeleton (scenes and sub-scenes), the higher *k* value enables reconstructing also geometrical models, and the highest *k* enables reconstructing the behavior scripts and therefore interacting with the complete scene. It encourages the community to gradually populate the host on which the scene is launched, since the “level of details” and the “level of interactivity” of the content increases as the number of content consumers grows.

3.4 Use Case Scenario for Virtual Museums

The solution proposed in the previous sections has been designed as a prerequisite to the new usage scheme for virtual museums that are deployed “in the cloud”, particularly on federated hosts constituting 3D virtual environment. Taking advantage of virtual environment functionality, virtual museum staff can build scenes containing multiuser educational scenarios with interactive 3D models. In the analyzed use case this content is secured, distributed and used in three phases: Content Securing Phase, Content Distribution Phase, and Content Reconstruction and Deployment Phase.

In the Content Securing Phase:

- Along with digital ticket to the virtual museum, the participants (not just visitors) obtain shares of scene decryption keys that are generated automatically;
- The scenes are dedicated exclusively for the target population of participants – shareholders;
- The shares generated for the given participant correspond to its credentials (user privileges, role, preferences, ticket class paid, etc);
- The protected scene subsets or models can be reused by content publishers as a building blocks of many different scenes released for distribution;

In the Content Distribution Phase:

- Participants having the shares roam and gather at freely chosen host, and therefore they “support” better hosts through the „wisdom of the crowd”;
- Additionally, the chance to have large number of users with their attractive content motivates host providers to assure quality of their service, particularly data security;
- Shareholders have ability to decide about the event context; it allows for decentralizing the content management, which is crucial in federated virtual environments, since their topology is decentralized by design;
- „Malicious host” cannot copy encrypted and shared scenes owned by users that are roaming through it (note that in case of regular Hypergrid model “malicious host” could copy the scene from the inventory of a single user who visited the host).

Finally, in the Content Reconstruction and Deployment Phase:

- Interaction with the scene is possible after virtual exhibition start which requires critical mass of at least k of n simultaneous participants (and additionally e.g. virtual guide with special shares);
- The usage scheme assumes, that even if the content leaks out after the keys are reconstructed and the scene is decrypted on the host, tickets (shares) already have been distributed in the number that makes the content creation cost-effective;
- Moreover, the exhibition with the shared scenes can be launched for a second time and subsequently, provided that k users will gather on any host. Obviously, the k number may include completely different set of users each time, so the content can reach broad range of consumers.

4 Conclusions and Future Work

In this paper the model for large scale content distribution, based on extended threshold secret sharing technique, is proposed, along with usage scheme, that bypasses the "malicious host" problem. Its main practical advantage is protection against illegal distribution of the scenes in federated virtual environment and, at the same time, providing the content consumers with freedom of deciding about the context of the interaction with the 3D scene. It opens the world of the federated virtual environments, its user communities and services, to the content providers such as virtual museums, assuring the desired level of security and cost-effectiveness (no content leaks, no piracy, large populations of users). The support for flexible and selective sub-scene protection is a key factor allowing for applying the proposed solution in practice.

In future the performance of the software implementation will be investigated more in detail. Share size influence the computation and communication efficiency, but since only encryption keys, not whole encrypted scenes, are shared in the proposed solution, splitting and reconstructing algorithms will not add any significant performance costs. However, computational effort related to scenes decryption on-the-fly needs to be tested.

An interesting aspect of secret sharing, that can be taken into account in the future research on the topic, is taking advantage of the knowledge about users rights and, at the same time, about inter-user and inter-role relations, hierarchies, and constraints (e.g. separation of duties), in the key generation process.

References

1. Cellary, W., Walczak, K.: Issues in Creation, Management, Search and Presentation of Interactive 3D Content. In: Interactive 3D Multimedia Content - Models for Creation, Management, Search and Presentation, pp. 37–54. Springer, London (2012)
2. Flotyński, J., Dalkowski, J., Walczak, K.: Building multi-platform 3D virtual museum exhibitions with Flex-VR. In: The 18th International Conference on Virtual Systems and Multimedia, pp. 391–398. IEEE Advancing Technology for Humanity, Milan (2012)
3. Eno, J., Gauch, S., Thompson, C.: Searching for the metaverse. In: Proc. of the 16th ACM Symp. on Virtual Reality Software and Technology, pp. 223–226. ACM, New York (2009)
4. Botev, J., Hohfeld, A., Schloss, H., Scholtes, I., Sturm, P., Esch, M.: The HyperVerse: concepts for a federated and Torrent-based '3D Web'. Int. J. Adv. Media Commun. 2(4), 331–350 (2008)
5. Hypergrid, `http://opensimulator.org/wiki/Hypergrid`
6. OpenSimulator, `http://opensimulator.org/`
7. Kokua/Imprudence, `http://blog.kokuaviewer.org/`
8. Horn, D., Cheslack-Postava, E., Azim, T., Freedman, M.J., Levis, P.: Scaling Virtual Worlds with a Physical Metaphor. IEEE Pervasive Computing 8(3), 50–54 (2009)
9. Alpcan, T., Bauckhage, C., Kotsovinos, E.: Towards 3D Internet: Why, What, and How? In: Proc. of the 2007 Int. Conf. on Cyberworlds, pp. 95–99. IEEE, Washington, DC (2007)
10. Sander, T., Tschudin, C.F.: Protecting Mobile Agents Against Malicious Hosts. In: Vigna, G. (ed.) Mobile Agents and Security. LNCS, vol. 1419, pp. 44–60. Springer, Heidelberg (1998)

11. Hypergrid Security, http://opensimulator.org/wiki/Hypergrid_Security
12. Hypergrid Ref. Guide, http://www.ics.uci.edu/~lopes/opensim/HypergridReferenceGuide.html
13. Lopes, C.: Hypergrid: Architecture and Protocol for Virtual World Interoperability. IEEE Internet Computing 15(5), 22–29 (2011)
14. Bierman, E., Cloete, E.: Classification of malicious host threats in mobile agent computing. In: Proc. SAICSIT, Republic of South Africa, pp. 141–148 (2002)
15. OSgrid, http://www.osgrid.org/
16. Open Cobalt, http://www.opencobalt.org/
17. Open Wonderland, http://openwonderland.org/
18. Bitmanagement BS Collaborate, http://www.bitmanagement.com/products/server/bs-collaborate/
19. Boneh, D., Sahai, A., Waters, B.: Functional encryption: Defnitions and challenges. Technical Report 2010/543, IACR ePrint Cryptography Archive (2010)
20. Shen, E., Shi, E., Waters, B.: Predicate privacy in encryption systems. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 457–473. Springer, Heidelberg (2009)
21. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: ACM Conference on Computer and Communications Security, pp. 89–98. ACM, New York (2006)
22. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
23. Goh, E.J.: Secure indexes. Technical Report 2003/216, IACR ePrint Cryptography Archive (2003)
24. Chase, M., Kamara, S.: Structured Encryption and Controlled Disclosure. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 577–594. Springer, Heidelberg (2010)
25. Song, D., Wagner, D., Perrig, A.: Practical techniques for searching on encrypted data. In: IEEE Symposium on Research in Security and Privacy, pp. 44–55. IEEE Computer Society, Washington, DC (2000)
26. Shamir, A.: How to Share a Secret. Communications of the ACM, 612–613 (1979)
27. Simmons, G.J.: An Introduction to Shared Secret and/or Shared Control Schemes and Their Application in Contemporary Cryptology. In: The Science of Information Integrity, pp. 441–497. IEEE Press (1992)
28. Tang, S.: Simple Secret Sharing and Threshold RSA Signature Schemes. Journal of Information and Computational Science 1, 259–262 (2004)
29. Ogiela, M.R., Ogiela, U.: Information Security Management Based on Grammar Threshold Schemes. In: 1st IEEE/IFIP International Workshop on Knowledge Management for Future Services and Networks, Osaka, pp. 247–250 (2010)
30. Wójtowicz, A.: Secure User-Contributed 3D Virtual Environments. In: Interactive 3D Multimedia Content – Models for Creation, Management, Search and Presentation, pp. 171–193. Springer, London (2012)
31. Wójtowicz, A., Cellary, W.: Representing User Privileges in Object-Oriented Virtual Reality Systems. In: Camarinha-Matos, L.M., Pereira, P., Ribeiro, L. (eds.) DoCEIS 2010. IFIP AICT, vol. 314, pp. 52–61. Springer, Heidelberg (2010)
32. Wójtowicz, A., Cellary, W.: Access Control Model for Dynamic VR Applications. In: Rea, A. (ed.) Security in Virtual Worlds, 3D Webs, and Immersive Environments: Models for Development, Interaction, and Management, pp. 284–305. IGI Global (2011)

33. Wójtowicz, A., Flotyński, J., Rumiński, D., Walczak, K.: Securing Learning Services Accessible with Adaptable User Interfaces. In: Grzech, A., Borzemski, L., Świątek, J., Wilimowska, Z. (eds.) Information Systems Archictecture and Technology. Service Oriented Networked Systems, pp. 109–118. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław (2011)
34. Wójtowicz, A., Walczak, K., Wiza, W., Rumiński, D.: Web Platform with Role-based Security for Decentralized Creation of Web 2.0 Learning Content. In: Proceedings of the 7th International Conference on Next Generation Web Services Practices (NWeSP), pp. 523–529. IEEE, Salamanca (2011)

Client-Side Detection of SQL Injection Attack

Hossain Shahriar, Sarah North, and Wei-Chuen Chen

Department of Computer Science
Kennesaw State University
Georgia, 30144, USA
{hshahria,snorth}@kennesaw.edu, wchen10@students.kennesaw.edu

Abstract. Despite the development of many server-side approaches, SQL Injection (SQLI) vulnerabilities are still widely reported. A complementary approach is to detect the attack from the client-side (browser). This paper presents a client-side approach to detect SQLI attacks. The client-side accepts shadow SQL queries from the server-side and checks any deviation between shadow queries with dynamic queries generated with user supplied inputs. We propose four conditional entropy metrics to measure the deviation between the shadow query and dynamic query. We evaluate the approach with an open source PHP application. The results indicate that our approach can detect malicious inputs early at the client-side.

Keywords: SQL Injection, conditional entropy, client-side attack, web security.

1 Introduction

The starting point of SQL Injection (SQLI) [1] attack is the client-side (browser). If malicious inputs can be detected at the browse, then many SQLI attacks could be thwarted by preventing them to be supplied to the server-side. This could bring two benefits: (i) adding as an extra protection layer on top of existing server-side solutions (secure coding [3], code generation [2], dynamic analysis [6]), and (ii) working as complementary to scanner tools [4, 5] that do not have the knowledge of intended SQL query structure.

There are challenges for developing client-side SQLI attack detection approach. For example, the relevant form fields that require checking at the client-side must be informed by the server-side and priori information about the query structure needs to be known. The supplied information from the server-side should not contain sensitive information such as the original names of query tables and columns. The alteration of dynamic query structure needs to be checked at the client-side.

This paper attempts to address these issues. We propose a server-side assisted client-side SQLI attack detection framework. We extract query structures from the server-side and convert them to shadow queries. A shadow query is identical to an original query, except column names, table names, and input variables are replaced with arbitrary symbolic values. We measure the deviation of information content between shadow queries and actual queries (replacing symbolic values with supplied

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 512–517, 2013.

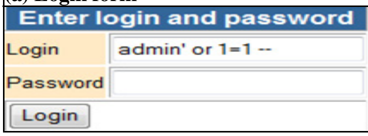
input values). We propose conditional entropy metrics (a form of measuring information content) as a basis to differentiate between benign and malicious queries at the client-side. If there is a deviation of the information content between a shadow query and a dynamic query (formed by replacing symbolic values with actual form field values), then an SQLI attack is detected at the client-side and a request is blocked. Otherwise, inputs supplied to a form are considered as benign and a request is allowed to the server-side. We evaluate our approach with an open source PHP web application containing known SQLI vulnerabilities. The results indicate that the approach can detect malicious attack inputs at the client-side.

The paper is organized as follows. Section 2 shows an example of SQLI attack followed by a brief discussion on the related works. In Section 3, the proposed approach is discussed in details. Section 4 discusses the experimental evaluation. Section 5 concludes the paper and discusses future work.

2 SQL Injection (SQLI) Attack and Client-Side Detection

We show a login form in Figure 1(a) with two input fields (*Login*, *Password*) whose values could contain malicious inputs for SQLI attacks. Figure 1(b) shows the HTML code of the form. When a user clicks on *Login* button, a request for accessing the website is performed with the supplied values. Here, the Login and Password inputs are accessed at the server-side through *$fLogin* and *$fPassword* variables. The *login.php* script is executed at the server-side which generates a dynamic SQL query to perform authentication based on the supplied values.

(a) Login form



(b) HTML of login form

```
<form name = “form1” action="login.php" method="post">
  <input type=”text” name= “fLogin”>
  <input type= “text” name= “fPassword”>
  <input type= “submit” name= “Login”>
</form>
```

Fig. 1. (a) A login form (b) HTML code of login form

```
1.  $login = $_POST[‘fLogin’];
2.  $pwd = $_POST[‘fPassword’];
3.  $qry = "select id, level from tlogin where uid =’”. $login. “’ and password =’”. $pwd. “’”;
4.  $result = mysql_query($qry);
5.  while($row = mysql_fetch_array($result)) { // authentication
6.      $_SESSION[’ID’] = $row['id'];
        ……..
7.  }
```

Fig. 2. PHP code for authentication

Figure 2 shows the PHP code snippet. Lines 1 and 2 extract from *fLogin* and *fPassword* fields of a request into *$login* and *$pwd* variables, respectively. The inputs are not filtered and concatenated with other strings for generating the dynamic SQL query at Line 3. Line 4 executes the query and Line 5 performs the authentication based on the presence of at least one row in the result set. If the provided credential information is matched, the current session id is set with the *id* field from the table.

If an attacker supplies malicious inputs as shown in Figure 1(a), the query becomes *select id, level from tlogin where uid ='' or 1=1 --' and password =''*. The resultant query is a tautology (the query after the comment symbol "--" is ignored). The remaining condition *uid='' or 1=1* is evaluated as *true*. Therefore, the malicious input alters the intended query structure implemented by a programmer.

(a) Revised HTML form

```
<form name = "form1" action="login.php" method="post">
  <input type="text" name= "fLogin">
  <input type= "text" name= "fPassword">
  <input type= "hidden" name= "sQuery"
    value= "select c1, c2 from t1 where c3 =v1 l1 c4 = v2">
  <input type= "hidden" name= "chkField" value ="fLogin,fPassword">
  <input type= "hidden" name= "substitue" value ="v1,v2">
   <input type= "submit" name= "Login" onclick= "chkSQLI ()">
</form>
```

(b) JavaScript code for checking SQLI attack

```
1.   function chkSQLI () {
2.     var sqry = document.form1.sQuery;
3.     var entropySqry = condEntropy(sqry);
4.     var aqry = subs (sqry, document.form1.chkField, document.form1.substitue);
5.     var entropyAqry = condEntropy(aqry);
6.     if (entropySqry == entropyAqry)
7.       document.form1.submit();
8.     else
9.     alert ("SQL injection attack inputs are present in supplied values");
10.  }
```

Fig. 3. (a) Modified HTML form (b) JavaScript code for checking SQLI attack

We now show how our approach can detect the attack from the client-side. We modify the HTML form as shown in Figure 3 (a) by adding three hidden fields. They represent server-side provided information of the shadow query (*sQuery*), the form fields that need to be checked (*chkField*) for malicious inputs, and the values in shadow query that need to be substituted with the form field values (*substitute*). A JavaScript code is supplied from the server-side to check SQLI attack inputs at the client-side. The method *chkSQLI* is shown in Figure 3(b). It extracts the shadow query in Line 2. Line 3 computes the conditional entropy of the shadow query (*entropySqry*). Line 4 substitutes shadow query's symbolic values (specified at *document.form1.substitue* as comma separated field values) with form fields (specified at *document.form1.chkField* as comma separated values). The conditional entropy (*entropyAqry*) of the actual query is obtained at Line 5. Line 6 compares the value of *entropyAqry* and *entropySqry*. If there is a match, the inputs are considered as benign and the form is submitted to the remote website (Lines 6-7). If there is a mismatch, a warning message is generated for SQLI attack related inputs (Line 9).

We now discuss about the framework briefly, server-side shadow query generation process, client-side conditional entropy computation technique in Section 3.

3 Client-Side SQLI Attack Detection Approach

First, server-side script is analyzed to identify HMTL forms that contain input fields. Then the SQL queries present in the script code is extracted and we find the relevant set of input fields in forms that contribute values during dynamic query generation process. The SQL query is simplified to generate a shadow query. This is done by replacing table, column, logical operator, and variable values with symbolic values. This step prevents revealing sensitive information at the client-side. The HTML forms are modified by adding hidden fields with the following: (i) the shadow query, (ii) fields relevant to dynamic queries and whose values need to be replaced with appropriate symbols in the shadow query, and (iii) the relevant symbols from shadow query that need to be replaced with user provided inputs.

The modified script is deployed in a web server. From a browser, when a user requests a web page to the server, the server-side returns the modified pages containing the revised HTML forms. When a user resubmits a form after providing inputs in form fields, the client-side checking takes place (JavaScript code). The checking is done by computing the conditional entropy of the shadow query and the actual query by substituting symbols with user provided input values. If any deviation is found, the inputs are flagged as malicious. The request is not forwarded to the server-side. Otherwise, the inputs are considered as benign and the request is forwarded to the server-side. We now briefly discuss about conditional entropy and its application for detecting client-side SQLI attacks.

Conditional entropy is a measurement technique from the area of information theory [7]. Given that we have a set of random variables, and we know the outcome of one variable in advance, the randomness of the remaining variables can be computed with the conditional entropy metric. For a given SQL query, if we have the priori information about the query type (select, insert), we can measure the information content about certain parts of the query. This allows us to reduce the computation time.

Let us assume that q be a query present in a program, $T=\{t_1, t_2, \ldots, t_M)$ be the set of tables being used in the query, $C=\{c_1, c_2, \ldots, c_L)$ be the set of all columns that can be selected, inserted, deleted, or updated in q, $V=\{v_1, v_2, \ldots, v_P)$ are the set of values that are used in where conditions or setting values, $L=\{l_1, l_2, \ldots, l_P)$ are the logical operators present in q, and $O=\{o_1, o_2, o_3, o_4\}$ which represents the *select* (o_1), *insert* (o_2), *update* (o_3), and *delete* (o_4) operation that can be performed in the query q.

We define four conditional probability functions as follows:

$P(c|o)$ is the conditional probability of column's presence given that we know the operation type of q, where $o \varepsilon O$.

$P(v|o)$ is the conditional probability of value v being used in a query's where condition or set values given that we know the operation type of q.

$P(t|o)$ is the conditional probability of table t being accessed or modified in a query given that we know the operation type of q.

$P(l|o)$ is the conditional probability of performing logical operation (in where condition) in q, given that we know the operation type of q.

We now define four conditional entropies of a query *q* as follows.

Given that we know the operation type of a query, the conditional entropy of column (denoted as H_c) can be computed as follows (*Equation (i)*):
$H_c(c|o) = -\sum_{c \varepsilon C} P(c,o) \log P(c|o) \ldots (i)$

H_c allows us to detect SQLI attacks where altered query selects additional columns (*e.g.*, *UNION select (1, 1)*) or reduces columns.

Given that we know the operation type of a query, the conditional entropy of values (denoted as H_v) can be computed as follows (*Equation (ii)*):
$H_v(v|o) = -\sum_{v \varepsilon V} P(v,o) \log P(v|o) \ldots (ii)$

H_v allows us to detect attacks where altered query may set new values to fields not intended by the original query (*e.g.*, tautology has 1=1 in attack signatures where 1 is assumed as a value).

Given that we know the operation type of a query, the conditional entropy of table (denoted as H_t) can be computed as follows (*Equation (iii)*):
$H_t(t|o) = -\sum_{t \varepsilon T} P(t,o) \log P(t|o) \ldots (iii)$

H_t allows us to detect attacks where altered query may perform additional operations on table not intended by the original query (*e.g.*, piggybacked queries selecting arbitrary tables).

Given that we know the operation type of a query, the conditional entropy of logical operation (denoted as H_l) can be computed as follows (*Equation (iv)*):
$H_l(l|o) = -\sum_{l \varepsilon L} P(l,o) \log P(l|o) \ldots (iv)$

H_l allows us to detect attacks where altered query may perform additional logical operations on table not intended by the original query (*e.g.*, tautology) as well as reduce the number of operators.

4 Evaluation

We evaluate the proposed approach using an open source PHP application named *PHP-Address book* (address and contact manger). We analyze 230 source files that have 38 forms, 157 input fields, 43 select, 11 update, 7 insert, and 7 delete queries.

We instrument the application with JavaScript code. We deploy the application in an Apache web server with MySQL server acting as the database. We then visit the application so that we can obtain the modified HTML forms. We supply malicious inputs (tautology, union, piggybacking) in each of the form fields. The approach detects the supplied malicious inputs in form fields.

5 Conclusions

This paper develops a client-side SQLI attack detection framework. Our approach relies on the generation of shadow queries as well as encoding information in HTML forms to enable the necessary checking at the client-side. We measure the deviation between expected query and altered query with four conditional entropy metrics. The approach detects malicious inputs causing SQLI at the server-side early and relieves the server-side for additional checking on SQLI attacks. We evaluate the proposed approach with an open source PHP application and our approach detects the presence of malicious inputs early at the client-side before being passed to the server-side.

Our future plan includes extending the approach for stored procedures, and evaluating our approach with more web applications. Moreover, we are planning to extend the concept for detecting other relevant attacks such as cross-site scripting. Currently, our approach focuses only SQLI attacks caused by providing malicious inputs in forms. So, we will work on identifying non-form input-based SQLI attack detection. Further, our goal also includes comparing the performance of client-side detection approach with other scanner tools.

Acknowledgement. This work was funded in part by Kennesaw State University College of Science and Mathematics Faculty Summer Research Award Program.

References

1. The Open Web Application Security Project (OWASP), `https://www.owasp.org/index.php/Top_10_2010-Main`
2. Johns, M., Beyerlein, C., Giesecke, R., Posegga, J.: Secure Code Generation for Web Applications. In: Massacci, F., Wallach, D., Zannone, N. (eds.) ESSoS 2010. LNCS, vol. 5965, pp. 96–113. Springer, Heidelberg (2010)
3. Antunes, N., Vieira, M.: Defending Against Web Application Vulnerabilities. IEEE Computer 45(2), 66–72 (2012)
4. W3af, Open Source Web Application Security Scanner, `http://w3af.org`
5. SQL-inject-me, `https://addons.mozilla.org/en-us/firefox/addon/sql-inject-me`
6. Agosta, G., Barenghi, A., Parata, A., Pelosi, G.: Automated Security Analysis of Dynamic Web Applications through Symbolic Code Execution. In: Proc. of the 9th International Conference on Information Technology: New Generations (ITNG), Las Vegas, NV, pp. 189–194 (April 2012)
7. Cover, T., Thomas, J.: Elements of Information Theory. John Wiley and Sons (2006)

Early Dealing with Evolving Risks in Long-Life Evolving Software Systems*

Le Minh Sang Tran

University of Trento, Italy
tran@disi.unitn.it

Abstract. Existing risk assessment methods often rely on a context of a target software system at a particular point in time. Such contexts of long-living software systems tend to evolve over time. Consequently, risks might also evolve. Therefore, in order to deal with evolving risks, decision makers need to select an appropriate risk countermeasure alternative that is more resilient to evolution than others. To facilitate such decision, we propose a pioneer method taking the uncertainty of evolutions and outputs of a risk assessment to produce additional information about the evolution resilience of countermeasure alternatives.

Keywords: risk assessment, evolving risk, max belief, deferral belief.

1 Introduction

Long-life software systems keep evolving to continuously satisfy changing business needs, new regulations, or the introduction of new technologies. Such evolutions might expose the software systems to new risks, and might make the output of the current risk analysis on the software systems become partially obsoleted. Consequently, the software systems might be no longer secure.

Risk assessment methods compatible with ISO 31000:2009 typically rely on a context of the target software system at a particular point of time. These methods help to identify risks and countermeasure alternatives (*i.e.* set of countermeasures). Decision makers then need to select the most appropriate alternative to implement to mitigate unacceptable risks. However, when the context evolves, risks might also evolve. Previously acceptable risks might become unacceptable or vice versa, or new risks emerge [1, Chap. 15]. For example, any risk mitigated by SHA-0 based countermeasure was acceptable before 2004, but might be unacceptable later since SHA-0 was efficiently attacked[1]. Thus, a current countermeasure alternative may no longer be appropriate and it is necessary to develop new ones to address the evolving or newly emerging risks. Obviously, implementing new ones to replace for obsoleted ones may be more expensive than having a one that still be appropriate. Decision makers then need to take into account the evolution of risks while selecting countermeasure alternative.

* This work is supported by the European Commission under the project EU-FP7-NoE-NESSOS.

[1] http://en.wikipedia.org/wiki/SHA-0#SHA-0, site visited on March, 2013.

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 518–523, 2013.

While there are lots of risk assessment methods *e.g.,* [1, 3–6, 9], few supports a systematic selection on risk countermeasure alternatives [4, 6, 9], and even less mentioning evolving risks [2]. Traditional risk assessment methods typically perform on a context at a particular point of time and hence could not guarantee the risk assessment results in an evolving context. Concerning to evolutions, in [2], the authors proposed a general technique and guideline for managing risk in changing systems. However, they did not mention the uncertainty of evolutions (*i.e.* likelihoods of occurrences), and how to use these information to support the decision making process.

The approach in this paper is an effort to fill some of that void. The focus of this paper is not on how to obtain the uncertainty information, but rather on how to make use of them to produce additional factors to support the decision making process. In particular, we propose a method to take the potential changes, their uncertainty, and risk assessment outputs to quantitatively evaluate the evolution-resilience of a countermeasure alternative.

2 Terminology

- *Context*: includes all required information to do risk assessment such as assumptions about the working environment, requirements model, targets to be protected and so on. It is the premises for and the background of the risk analysis, as well as the purposes of the analysis and to whom the risk analysis is addressed [1, Chap. 5]. According to ISO 31000:2009, a context includes all external factors (*e.g.,* regulatory, environment) and internal factors (*e.g.,* business process, policies, standards, system functions, reference models).
- *Before context*: is the current context at the current time.
- *After context*: is the future context with potential changes.

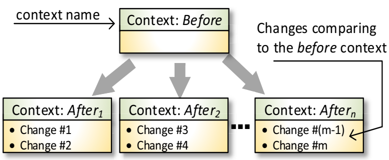
3 The Proposed Method

Our proposed method includes following steps:

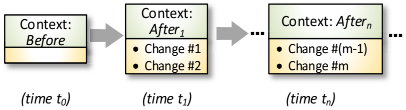
- Step 1 *Identify evolving contexts:* identify the current context and all its possible evolutions.
- Step 2 *Perform risk assessment:* do risk assessment on identified contexts.
- Step 3 *Model context evolution:* describe the context evolution model based on identified contexts and their corresponding risk assessment outputs.
- Step 4 *Perform evolution analysis:* perform analysis on the context evolution model to calculate the quantitative evolution metrics to support the selection of an appropriate countermeasure alternative to address risks.

3.1 Step 1 – Identify Evolving Contexts

This step takes all documents about the planned and potential changes of the system as inputs. We consider four different evolution perspectives: *maintenance,*



(a) Before-after evolution



(b) Continuous evolution

Fig. 1. The evolution perspectives of contexts

before-after, *continuous evolution*, and *hybrid evolution*. The first three perspectives were discussed in [1, Chap. 15]. The *maintenance* perspective relates to the outdate of a risk document of an existing system. Hence it is not the focus of this work. The *before-after* perspective predicts future contexts by anticipating planned and unplanned changes in the current context. The *continuous* evolution perspective predicts the evolution of the current context over time based on planned gradual changes. The *hybrid* evolution perspective is the combination of *before-after* and *continuous* ones where both planned and unplanned changes are considered in a sequence of time.

We abuse the notation of *before* and *after* contexts to represent these evolution perspectives (except the *maintenance* one). Fig. 1(a) demonstrates the *before-after* evolution perspective. A context is depicted as a rectangle with child compartments. The first compartment shows the context name, and the second compartments exhibits the changes comparing to the *before* context. In this perspective, a *before* context might have many possibilities to evolve to other *after* contexts, denoted as *evolution possibility*. At the end of the day, exact one possibility materializes. Each evolution possibility associates with an *evolution probability* which is the likelihood that a possibility materializes. This uncertainty is because "the only certainty is that nothing is certain"(*Pliny the Elder*[2]). Fig. 1(b) illustrates the *continuous* evolution perspective where changes happen continuously. The *before* context at current time t_0 might evolve an *after* context at time t_1 which might continuously evolve at time t_2, and so forth.

After contexts can be identified by using any input document that describes potential changes (either planned or unplanned) in the current context. Unplanned changes could be anticipated by domain experts by using several techniques such as brainstorming with chalk and blackboard, or techniques for requirements changes anticipation. Readers are referred to [10, Chap. 6] for a more detailed discussion of these techniques. The evolution probabilities are the experts' belief that evolution possibilities might happen. The probability semantic is accounted by using the game-theoretic approach described in [8].

3.2 Step 2 – Perform Risk Assessment

In this step, we employ a state-of-the-art risk assessment method (*e.g.,* Attack Trees [5], Cause-Consequence Diagrams [3], and CORAS [1]) to perform risk

[2] Gaius Plinius Secundus (23 – 79), a Roman naturalist, and natural philosopher.

assessment for identified contexts. The outcome of this step is list of risk countermeasure alternatives, which are also the output of a risk assessment method.

A *risk countermeasure alternative* includes a list of countermeasures, and the residual risks (with residual risk level) of a system after implementing the countermeasures. A countermeasure could be a security controls (*e.g.,* technology, policy), or a high level security requirement that mitigates risks. A risk level is a pair of the likelihood by which a risk might occur, and its impact. Based on risk level, a risk is categorized, such as *acceptable* or *unacceptable*. A residual risk level is the risk level after implementing countermeasures.

When performing risk assessment on *after* contexts, we can do either a full risk assessment from scratch, or an incremental risk assessment taking advantage on the risk assessment on the *before* context. Needless to say, the former strategy does not use resources efficiently. The latter is better since it only addresses the changed parts of the *after* context comparing to the *before* context [1, Chap.15].

3.3 Step 3 – Model Context Evolution

This step takes the identified contexts and their corresponding risk countermeasure alternatives to establish the context evolution model. We employ the approach described in [8] to model the context evolution in terms of evolution rules. There are two kinds of rules: *observable rule* and *controllable rule*. The former captures the way how the context evolves. The latter captures different alternatives to address risks in each context. An evolved context, as aforementioned, is foreseen with a certain evolution probability. To the sake of simplicity, we assume that the evolving contexts identified in Step 1 are complete and mutual exclusive. In other words, exact one of the *after* contexts materializes at the end.

Let $\mathcal{C}$ be an context, and $\mathcal{C}_i$ be the i^{th} *after* context of $\mathcal{C}$, and CA_j be a risk countermeasure alternative of $\mathcal{C}$. The observable rule (r_o) and controllable rules (r_c) are described as follows.

$$r_o(\mathcal{C}) = \left\{ \mathcal{C} \xrightarrow{p_i} \mathcal{C}_i \,\middle|\, \sum_{i=1}^{n} p_i = 1 \right\} \tag{1}$$

$$r_c(\mathcal{C}) = \{\mathcal{C} \rightarrow CA_j \,|j = 1..m\} \tag{2}$$

where n is the number of *after* contexts of $\mathcal{C}$; p_i is the evolution probability for which $\mathcal{C}$ evolves to $\mathcal{C}_i$; m is the number of risk countermeasure alternatives of $\mathcal{C}$. The sum of all p_i is 1 since the *after* contexts are complete and mutual exclusive.

The *before-after* evolution perspective is represented by an observable rule. The *continuous* and *hybrid* evolution perspectives are represented as a sequence of observable rules where the current context of an observable rule is the *after* context of another observable rule, so on and so forth.

Fig. 2 shows a graphical visualization of the context evolution model of the *hybrid* evolution perspective. The observable rule is denoted by connections from a *before* context to *after* contexts. The decorators on the connections are the evolution probabilities. To denote the controllable rule, the rectangles representing

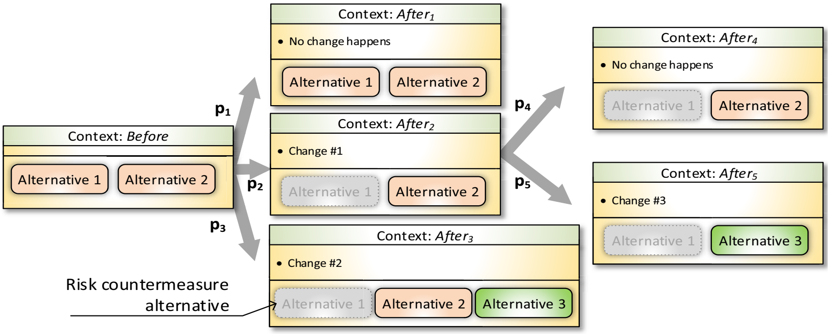


Fig. 2. The context evolution model

context are extended with a new compartment containing risk countermeasure alternatives which are represented by round rectangles. The controllable rule then is understood as different risk countermeasure alternatives of a context.

3.4 Step 4 – Perform Evolution Analysis

This step performs an evolution analysis on the context evolution model to calculate evolution metrics that support the decision making process. In particular, the evolution metrics aim to answer the question that to what extent a risk countermeasure alternative can resist the evolution. This analysis relies on two quantitative metrics: *max belief* and *deferral belief* [8][3].

Max Belief (*MaxB*): is the maximum belief that a risk countermeasure alternative will be appropriate if evolutions happen. By term *appropriate*, we mean the residual risks after applying the countermeasure alternative in the evolved contexts will still be acceptable. So, the system will still be safe.

Deferral Belief (*DefB*): is the belief that a risk countermeasure alternative will be inappropriate after evolutions happen. It is also the belief by which the implementation of the risk countermeasure alternative should be delayed until the context is clearly known.

We define a binary function $\mathsf{appropriate}()$ that takes two inputs: a context $\mathcal{C}$, and a risk countermeasure alternative CA, to produce 1 if CA is appropriate within $\mathcal{C}$, or 0 otherwise. The *max belief* and *deferral belief* of CA for the *before-after* evolution of the context $\mathcal{C}$ are as follows.

$$MaxB(CA|_{\mathcal{C}}) = \max_{\{\langle \mathcal{C} \xrightarrow{p_i} \mathcal{C}_i \rangle \in r_o(\mathcal{C}) | \mathsf{appropriate}(\mathcal{C}_i, CA)\}} p_i \tag{3}$$

$$DefB(CA|_{\mathcal{C}}) = 1 - \sum_{\{\langle \mathcal{C} \xrightarrow{p_i} \mathcal{C}_i \rangle \in r_o(\mathcal{C}) | \mathsf{appropriate}(\mathcal{C}_i, CA)\}} p_i \tag{4}$$

To the perspective of evolution-resilience, a better alternative is one that has a higher *max belief* and a lower *deferral belief*.

[3] We rename the *Residual Risk* metric in [8] to *deferral belief* to avoid naming conflict.

4 Conclusion and Future Work

The context in long-life evolving software systems might evolve over time. As the result, the risks of software systems might also evolve. Under the evolution of risks, the systems might be no longer secure. Therefore, new countermeasures need to be implemented to mitigate new risks. This however is much more expensive than addressing these risks at development time.

We are aware of only one study in the field [2] that introduced general techniques and guidelines for managing risk in evolving systems, but did not mention to the uncertainty of evolutions. Our work is pioneer in filling in the gap by proposing a method inspired from metrics in [8] to evaluate the evolution-resilience of the risk countermeasure alternatives which are the outputs of risk assessment. This provides more insights about the evolving risks to support decision makers in selecting the most appropriate alternative. We refer interested readers to [7] for a more detail discussion.

In future, we plan to extend the proposed method to provide quantitative metrics combining the *max belief*, and the *deferral belief* with the risk assessment output to facilitate the selection of countermeasure alternative. A promising approach is to employ the *Overall Cost* metric described in [9]. Also as a part of future work, we aim to develop a technique exploiting these quantitative metrics to suggest the countermeasure alternative that efficiently address the evolving risks with (or without) an optimal cost.

References

1. Lund, M.S., Solhaug, B., Stølen, K.: Model-Driven Risk Analysis. The CORAS Approach. Springer (2011)
2. Lund, M.S., Solhaug, B., Stølen, K.: Risk analysis of changing and evolving systems using CORAS. In: Aldini, A., Gorrieri, R. (eds.) FOSAD 2011. LNCS, vol. 6858, pp. 231–274. Springer, Heidelberg (2011)
3. Mannan, S., Lees, F.: Lees' Loss Prevention in the Process Industries, 3rd edn., vol. 1. Butterworth-Heinemann (2005)
4. Norman, T.L.: Risk Analysis and Security Countermeasure Selection. CRC Press, Taylor & Francis Group (2010)
5. Schneier, B.: Attack Trees: Modeling Security Threats. Dr. Dobbs Journal of Software Tools 24(12), 21–29
6. Stoneburner, G., Goguen, A., Feringa, A.: Risk Management Guide for Information Technology Systems. Tech. report, NIST (2002)
7. Tran, L.M.S.: Early Dealing with Evolving Risk in Long-life Evolving Software Systems. Tech. report, University of Trento (2013), `http://disi.unitn.it/~tran`
8. Tran, L.M.S., Massacci, F.: Dealing with Known Unknowns: Towards a Game-Theoretic Foundation for Software Requirement Evolution. In: Mouratidis, H., Rolland, C. (eds.) CAiSE 2011. LNCS, vol. 6741, pp. 62–76. Springer, Heidelberg (2011)
9. Tran, L.M.S., Solhaug, B., Stølen, K.: An Approach to Select Cost-effective Risk Countermeasures Exemplified in CORAS. CoRR (2013), `http://arxiv.org/abs/1302.4689`
10. van Lamsweerde, A.: Requirements Engineering - From System Goals to UML Models to Software Specifications. Wiley (2009)

Modeling and Analyzing Information Integrity in Safety Critical Systems

Mohamad Gharib and Paolo Giorgini

DISI, University of Trento,
Italy, Trento, Via Sommarive 14
gharib@disi.unitn.it, paolo.giorgini@unitn.it

Abstract. Preserving information integrity represent an urgent need for safety critical systems, where depending on incorrect or inconsistent information may leads to disasters. Typically, information integrity is a problem handled at technical level (e.g., checksumming). However, information integrity has to be analyzed in the social-technical context of the system, since information integrity related problems might manifest themselves in the business processes and actors interactions. In this paper, we propose an extended version of *i**/ secure Tropos modeling languages to capture information integrity requirements. We illustrate the Datalog formalization of the proposed concepts and analysis techniques to support the analyst in the verification of integrity related properties. Air Traffic Management (ATM) case study is used throughout the paper.

Keywords: Integrity, Security, Modeling, Analysing.

1 Introduction

Information integrity is surely one essential feature of any safety critical system, where incorrect or inconsistent information may produce disaster and loss of humans lives. Information integrity is traditionally considered in the way information are stored and transmitted, several techniques and solutions have been proposed (e.g., checksum [6]). However, such solutions do not solve problems that may rise at business and organizational level. So for example, in the Chicago-O'Hare runway collision [10], the *Air Traffic Controller* (ATCs) failed to notify the *Airplane Captain* that one of the runways was being used by another airplane, which results in inconsistent information within the system that lead to the runway collision, such problem cannot be handled by the available technical solutions since the problem (inconsistency) manifested itself in the loss business process.

Information integrity should be always considered as a socio-technical problem, where socio and organizational aspects of the system have to be considered along with the technical solutions, i.e., the critical system design has always to go through a socio-technical analysis, where needs about information integrity have to be identified with respect to all involved social and technical actors, their local and global objectives and their mutual interactions. In this paper,

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 524–529, 2013.

we extend $i*$/ secure Tropos modeling languages with the required concepts and primitives that enable us to capture and analyze the information integrity requirements. The paper is organized as follows, ATM case study in section (§2). Section (§3) discusses modeling information integrity requirements, while section (§4) presents the Datalog formalization. Section (§5) discuss the properties of the design, related work in section (§6), finally, conclusion and future work are presented at section (§7).

2 Case Study

Our case study concerns Air Traffic Management (ATM), which is a safety-critical system since its failures may result in loss of humans lives. An ATM system is managed by Air Traffic Control Center (ACC) that provides air traffic control (ATC) services, which are provided by ground-based controllers (ATCs) working at different Air Traffic Control Unit (ATCU) with the main purpose of preventing aircraft collisions by providing airplanes with the required separation information. Information handled by ATM, includes but not limited to, flight plans, separation data, and meteo data. For instance, any airlines company want to fly in Europe has to send its flight plan to EUROCONTROL where it is checked (might be modified) and send out to all ATCU that is affected by the flight. Furthermore ATCs, exchange information concerning the traffic flow in their sectors, and they have to provide airplanes with separation data during the different phases of the flight.

In such systems, information (e.g., separation data) might be used by several actors (e.g., ATCs) and some of those actors might modify it for different reasons. The system should be able to maintain (preserve) the integrity of information among all its users at any time to prevent any disasters that might happen due to information integrity related problems. For example we show how modification permissions should be delegated based on some criteria that can be captured only by the social aspects of the system (e.g., trust).

3 Modeling Information Integrity Requirements

In this section, we discuss information[1] and its integrity dimensions to get better understanding of what information integrity requirements are and when they are needed. Information can be produced in several ways by information producers that represent its initial source, and we call the actor(s) who consume(s) information as information consumer(s). For example, *Airline company* is the producer of "flight plan", and *Airplane captain* is the "flight plan" consumer. While information integrity is defined as the representational faithfulness of information to the true state of the object that the information represents [3]. Information integrity is a multi-dimensional concept [4,3] that can be characterized using multiple dimensions: accuracy, completeness and consistency. While integrity requirements means preserving these 3 dimensions.

[1] In this paper data and information are used as synonyms.

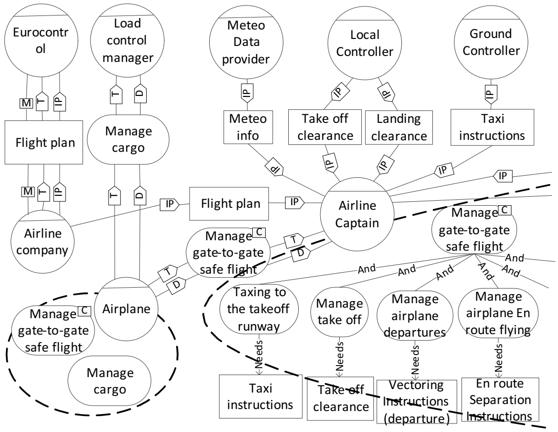


Fig. 1. A partial goal model concerning the ATM scenario

Within the organizational context not all the goals have the same criticality, compromised information might be a main reason for a goal failure and preserving information integrity is not free. The information integrity requirements will be determined based on the criticality of the consuming goal. Thus, we introduce two new concepts, namely, critical goal and critical information. The first is used to represent the stakeholders critical goals, while the second is information consumed by critical goals and its integrity should be preserved at any given time. For example, "manage gate-to-gate safe flight" is a critical goal and it is decomposed into several sub-goals, each one of them is a critical goal and consumes critical information.

Information might lose its integrity during the provision process. Thus, beside the normal provision (P), we define integrity provision (I-Provision) that can preserve the integrity of the provided information. Furthermore, we discuss the effect of trust over goals and permissions delegation, since it represents the mental counter-part of delegation [5]. For example, *Airlines company* delegates modification permission concerning "flight plan" to *EUROCONTROL* a trust relation has to exist to guarantee that the integrity of "flight plan" will be preserve. Figure 1 shows a partial goal model concerning the ATM scenario, in which the *Airline company* provide (I-Provision) " flight plan" to both *EUROCONTROL* and the *Airline captain*, and delegates modification permission over it along with a trust relation to *EUROCONTROL*. *Airplane* has 2 goals, it delegates them along with trust relation to *Load control manager* and *Airline captain*. "Manage gate-to-gate safe flight" is critical and it is decomposed into several sub-goals each of them is critical as well.

Table 1. General predicate

$needs(needer : n, info : i)$, $needer \in \{actor, goal\}$	$wants(actor : a, goal : g)$
$produce(goal : g, info : i)$	$modify(goal : g, info : i)$
$has(actor : a, info : i)$	$producer(actor : a, info : i)$
$consumer(actor : a, info : i)$	$provide(actor : a, actor : b, info : i)$
$prvChain(actor : a, actor : b, info : i)$	$ip\_provide(actor : a, actor : b, info : i)$
$ip\_prvChain(actor : a, actor : b, info : i)$	$can\_prv(actor : a, info : i)$
$can\_sat(actor : a, goal : g)$	$is\_responsible(actor : a, goal : g)$
$and\_dec(goal : g, goal : \mathrm{g}_1, goal : \mathrm{g}_2)$	$or\_dec(goal : g, goal : \mathrm{g}_1, goal : \mathrm{g}_2)$
$delegate(actor : a, actor : b, goal : g)$	$delChain(actor : a, actor : b, goal : g)$
$critical\_goal(goal : g)$	$critical\_info(info : i)$
$critical\_consumer(actor : a, info : i)$	$need\_perm(type : t, actor : a, info : i)$
$have\_perm(type : t, actor : a, info : i)$	$grant\_perm(type : t, actor : a, actor : b, info : i)$
$have\_grant\_perm(type : t, actor : a, info : i)$	$grant\_grant\_perm(type : t, actor : a, actor : b, info : i)$
$trust(actor : a, actor : b, goal : g)$	$tChian(actor : a, actor : b, goal : g)$
$trust\_perm(type : t, actor : a, actor : b, info : i)$	$trust\_pChian(type : t, actor : a, actor : b, info : i)$
$trust\_grant\_perm(type : t, actor : a, actor : b, info : i)$	$trust\_grant\_pChain(type : t, actor : a, actor : b, info : i)$
$satisfied(actor : a, goal : g)$	$consumed(actor : a, info : i)$
$critical\_consumed(actor : a, info : i)$	$integ\_preserved(actor : a, info : i)$
$integ\_comp(actor : a, info : i)$	

4 Formalizing Information Integrity Requirements

We use Datalog [1] as the underlying semantic framework. Table 1 introduces the general predicates. Table 2 lists the general axioms. For example, O1 expresses how a goal needs became an actors need, D1-2 for goal delegation, C3-5 an actor capabilities of having information, and C6-8 information provision capabilities, while C14-18 goals satisfying capabilities, CG1-2 criticality propagation and CG3-4 critical information and consumer. In table 3, P1 for having permission, P2-3 for granting permission. T1-2 trust concerning a goal satisfying. A1-4 actor believes its goal will be satisfied, and A5-6 consumer believes its information will be provided. A7 actor believes integrity is preserved, A8-10 where integrity might be compromised.

Table 2. General axioms

O1 $needs(A, I) \leftarrow is\_responsible(A, G) \wedge needs(G, I)$	O2 $wants(A, G) \leftarrow delChain(B, A, G)$
O3 $wants(A, G_{1[2]}) \leftarrow and\_dec(G, G_1, G_2) \wedge wants(A, G)$	O4 $wants(A, G_{1[2]}) \leftarrow or\_dec(G, G_1, G_2) \wedge wants(A, G)$
C1 $producer(A, I) \leftarrow can\_sat(A, G) \wedge produce(G, I)$	C2 $consumer(A, I) \leftarrow needs(A, I)$
C3 $has(A, I) \leftarrow producer(A, I)$	C4 $has(A, I) \leftarrow provide(B, A, I) \wedge can\_prv(B, I)$
C5 $has(A, I) \leftarrow ip\_provide(B, A, I) \wedge can\_prv(B, I)$	C6 $can\_prv(A, I) \leftarrow has(A, I)$
C7 $can\_prv(A, I) \leftarrow prvChain(A, B, I) \wedge has(B, I)$	C8 $can\_prv(A, I) \leftarrow ip\_prvChain(A, B, I) \wedge has(B, I)$
C9 $prvChain(A, B, I) \leftarrow provide(A, B, I)$	C10 $prvChain(A, B, I) \leftarrow \begin{cases} prvChain(A, C, I) \\ \wedge\, prvChain(C, B, I) \end{cases}$
C11 $ip\_prvChain(A, B, I) \leftarrow ip\_provide(A, B, I)$	C12 $ip\_prvChain(A, B, I) \leftarrow \begin{cases} ip\_prvChain(A, C, I) \\ \wedge\, ip\_prvChain(C, B, I) \end{cases}$
C13 $is\_responsible(A, G) \leftarrow wants(A, G) \wedge can\_sat(A, G)$	C14 $can\_sat(A, G) \leftarrow delChain(A, B, G) \wedge can\_sat(B, G)$
C15 $can\_sat(A, G) \leftarrow needs(G, I) \wedge has(A, I)$	C16 $can\_sat(A, G) \leftarrow \begin{cases} and\_dec(G, G_1, G_2) \wedge \\ can\_sat(A, G_1) \wedge can\_sat(A, G_2) \end{cases}$
C17 $can\_sat(A, G) \leftarrow or\_dec(G, G_1, G_2) \wedge can\_sat(A, G_1)$	C18 $can\_sat(A, G) \leftarrow or\_dec(G, G_1, G_2) \wedge can\_sat(A, G_2)$
D1 $delChain(A, B, G) \leftarrow delegate(A, B, G)$	D2 $delChain(A, B, G) \leftarrow \begin{cases} delChain(A, C, G) \\ \wedge\, delChain(C, B, G) \end{cases}$
CG1 $critical\_goal(G_{1[2]}) \leftarrow \begin{cases} and\_decom(G, G_1, G_2) \\ \wedge\, critical\_goal(G) \end{cases}$	CG2 $critical\_goal(G_{1[2]}) \leftarrow \begin{cases} or\_decom(G, G_1, G_2) \\ \wedge\, critical\_goal(G) \end{cases}$
CG3 $critical\_info(I) \leftarrow \begin{cases} needs(G, I) \wedge \\ critical\_goal(G) \end{cases}$	CG4 $critical\_cunsumer(A, I) \leftarrow \begin{cases} is\_responsible(A, G) \wedge \\ critical\_goal(G) \wedge needs(G, I) \end{cases}$

Table 3. Permission, Trust axioms and Achieved Actors Objectives

P1 $have\_perm(T,A,I) \leftarrow \begin{cases} grant\_perm(T,A,B,I) \\ \wedge\, have\_grant\_perm(T,A,I) \end{cases}$	P2 $have\_grant\_perm(T,A,I) \leftarrow producer(A,I)$
P3 $have\_grant\_perm(T,A,I) \leftarrow \begin{cases} grant\_grant\_perm(T,B,A,I) \\ \wedge\, have\_grant\_perm(T,B,I) \end{cases}$	P4 $need\_perm(T,A,I) \leftarrow \begin{cases} is\_responsible(A,G) \\ \wedge\, modify(G,I) \end{cases}$
T1 $tChain(A,B,G) \leftarrow trust(A,B,G)$	T2 $tChain(A,B,G) \leftarrow \begin{cases} tChain(A,C,G) \\ \wedge\, tChain(C,B,G) \end{cases}$
T3 $trust\_pChain(T,A,B,I) \leftarrow trust\_perm(T,A,B,I)$	T4 $trust\_pChain(T,A,B,I) \leftarrow \begin{cases} trust\_pChain(T,A,C,I) \\ \wedge\, trust\_pChain(T,C,B,I) \end{cases}$
T5 $trust\_grant\_pChain(T,A,B,I) \leftarrow trust\_grant\_perm(T,A,B,I)$	T6 $trust\_grant\_pChain(T,A,B,I) \leftarrow \begin{cases} trust\_grant\_pChain(T,A,C,I) \\ \wedge\, trust\_grant\_pChain(T,C,B,I) \end{cases}$
A1 $satisfied(A,G) \leftarrow is\_responsible(A,G)$	A2 $satisfied(A,G) \leftarrow \begin{cases} delChain(A,B,G) \wedge tChain(A,B,G) \\ \wedge\, satisfied(B,G) \end{cases}$
A3 $satisfied(A,G) \leftarrow \begin{cases} and\_dec(G,G_1,G_2) \wedge \\ satisfied(A,G_1) \wedge satisfied(A,G_2) \end{cases}$	A4 $satisfied(A,G) \leftarrow or\_dec(G,G_1,G_2) \wedge satisfied(A,G_{1[2]})$
A5 $consumed(A,I) \leftarrow has(A,I)$	A6 $consumed(A,I) \leftarrow prvChain(A,B,I) \wedge can\_prv(B,I)$
A7 $integ\_preserved(A,I) \leftarrow \begin{cases} critical\_cunsumer(A,I) \\ \wedge\, not\, integ\_comp(A,I) \end{cases}$	A8 $integ\_comp(A,I) \leftarrow prvChain(B,A,I)$
A9 $integ\_comp(A,I) \leftarrow \begin{cases} have\_perm(modify,B,I) \\ \wedge\, trust\_perm(modify,A,B,I) \end{cases}$	A10 $integ\_comp(A,I) \leftarrow \begin{cases} have\_grant\_perm(modify,B,I) \\ \wedge\, trust\_grant\_perm(modify,A,B,I) \end{cases}$

Table 4. Properties of the Design

Pro 1 :- $provide(B,A,I), not\, can\_prv(B,I)$	Pro 2 :- $provide(B,A,I), not\, wants(A,I)$
Pro 3 :- $delChain(A,B,G), not\, tChain(A,B,G)$	Pro 4 :- $need\_perm(T,A,I), not\, have\_perm(T,A,I)$
Pro 5 :- $have\_perm(T,A,I), not\, need\_perm(T,A,I)$	Pro 6 :- $need\_perm, not\, have\_grant\_perm(T,A,I)$
Pro 7 :- $have\_grant\_perm, not\, need\_perm(T,A,I)$	Pro 8 :- $critical\_cunsumer(A,I), not\, integ\_preserved(A,I)$

5 Analyzing Information Integrity Requirements

We use the DLV system[2] to analyze information integrity requirements, we define a set of properties that is used to verify the correctness of the model. In table 4, Pro1 allows the model to detect any invalid information provision / chain, Pro2 allows the model to detect any unneeded information provision that may threaten information integrity. For example, *Airlines company* does not have a provision capability concerning meteo information (Pro1), since it is not a meteo producer and such information will not be provided to it (Pro2). Depending on Pro3 the model will detect any situation in which there is no valid trust chain along with a goal delegation chain, which may endanger the satisfaction of the delegated goal. For example, *Airplane* delegates "Manage gate-to-gate safe flight" to *Airplane captain*, the model is able to detect and notify the designer if there is no trust chain concerning this goal delegation. According to Pro 4 permissions should be delegated only to actors require them, and they should not be delegated unless the actors require them (Pro 5). For example, *Airline captain* should not have modify permission over "flight plan", and *EUROCONTROL* should have such permission. Similarly, Pro 6-7 are specialized for grant permissions. Finally, Pro 8 enables the model to detect situations that might compromise critical information integrity. For example, the *Airline captain* believes that "flight plan" integrity is preserved, if it was not compromised by situations listed in table 3 (A8-10).

[2] `http://www.dbai.tuwien.ac.at/proj/dlv`

6 Related Work

Requirements engineering community did not appropriately support the modeling and analyzing of information integrity requirements. For instance, abuse frames [8] addresses the integrity problem by preventing unauthorized actors from modifying information. In both UMLsec and SecureUML [7,2] integrity was modeled as constraints that can restrict unwanted modification. The main limitation in these languages is that they do not consider the social relation among actors of the system (e.g. delegation, trust). Secure Tropos [9] provides concepts for capturing security requirements (mainly privacy) but offers no primitives neither to capture nor analyze information integrity requirements.

7 Conclusions and Future Work

We extend i */ secure Tropos [11,9] with several concepts including critical information, critical goal, information producer and consumer. The first two concepts are used to determine were integrity requirements are needed; the third is used to control modification permission properly by information producer, while consumer is used to determine if integrity has been preserved at its final destination. The extended framework with all new concepts presented in this paper will be supported by RE-Tool. Furthermore, we will extend the language with concepts and primitives for capturing the related information integrity dimensions (accuracy, completeness and consistency).

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of databases, Citeseer, vol. 8 (1995)
2. Basin, D., Doser, J., Lodderstedt, T.: Model driven security: From UML models to access control infrastructures (TOSEM) 15, 39–91 (2006)
3. Boritz, J.: IS practitioners' views on core concepts of information integrity. International Journal of Accounting Information Systems 6, 260–279 (2005)
4. Bovee, M., Srivastava, R., Mak, B.: A conceptual framework and belief-function approach to assessing overall information quality. International Journal of Intelligent Systems, Wiley Online Library 18, 51–74 (2003)
5. Castelfranchi, C., Falcone, R.: Principles of trust for MAS: Cognitive anatomy, social importance, and quantification. Multi Agent Systems (1998)
6. Cohen, F.: A cryptographic checksum for integrity protection Computers & Security. Elsevier 6, 505–510 (1987)
7. Jrjens, J.: Secure systems development with UML. Springer (2005)
8. Lin, L., Nuseibeh, B., Ince, D., Jackson, M., Moffett, J.: Introducing abuse frames for analysing security requirements. IEEE Computer Society (2003)
9. Mouratidis, H., Giorgini, P.: Secure Tropos: A security-oriented extension of the Tropos methodology. World Scientific Publishing, Singapore (2007)
10. safety board null, N. T. Aircraft accident report File No. 1-0017 (1972)
11. Yu, E.S.-K.: Modelling strategic relationships for process reengineering. Ph.D. thesis, University of Toronto (1996)

Author Index