LNBIP 112

Marko Bajec
Johann Eder (Eds.)

Advanced Information Systems Engineering Workshops

CAiSE 2012 International Workshops
Gdańsk, Poland, June 2012
Proceedings

Springer

Lecture Notes in Business Information Processing 112

Series Editors

Marko Bajec
Johann Eder (Eds.)

Advanced Information Systems Engineering Workshops

CAiSE 2012 International Workshops
Gdańsk, Poland, June 25-26, 2012
Proceedings

Springer

Volume Editors

Marko Bajec
University of Ljubljana
Faculty of Computer and Information Science
Laboratory for Data Technologies
Ljubljana, Slovenija
E-mail: marko.bajec@fri.uni-lj.si

Johann Eder
Alpen Adria Universität Klagenfurt
Institut für Informatik-Systeme
Klagenfurt, Austria
E-mail: johann.eder@aau.at

ISSN 1865-1348 e-ISSN 1865-1356
ISBN 978-3-642-31068-3 e-ISBN 978-3-642-31069-0
DOI 10.1007/978-3-642-31069-0
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2012939338

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

CAiSE is a well-established highly visible conference series on information systems engineering. CAiSE conferences also have the tradition of hosting several workshops which focus on particular topics in the field and provide ample room for discussion of new developments.

This year, CAiSE hosted ten workshops that were carefully selected based on the quality and maturity of their proposals, their potential and attractiveness, their fit to the information systems engineering field, and finally, their history. The papers accepted in eight of these workshops are contained in these proceedings. These are (in reverse alphabetical order):

- WISSE: Workshop on Information Systems Security Engineering
- ONTOSE: Workshop on Ontology, Models, Conceptualization and Epistemology in Social, Artificial and Natural Systems
- IWSSA: Workshop on System/Software Architectures
- HC-PAIS: Workshop on Human-Centric Process-Aware Information Systems
- GRCIS: Workshop on Governance, Risk and Compliance in Information Systems
- EOMAS: Workshop on Enterprise and Organizational Modeling and Simulation
- BUSITAL: Workshop on Businness/IT Alignment and Interoperability
- AgilES: Workshop on Agility of Enterprise Systems

Some statistics: the workshops in these proceedings attracted a total of 104 submissions. After a thorough review process managed by the respective workshop organizers, 35 full papers were accepted (34%) and 17 short papers. The overall acceptance rate for full papers is 34% and fur full+short papers 50%.

Two further workshops organized individual proceedings published elsewhere:

- CSSI: Workshop on Case Studies in Service Innovation
- NGEBIS: Workshop on Next-Generation Enterprise and Business Innovation Systems

The variety of topics addressed in this book confirm the fact that although information systems engineering is a well-established and mature discipline it is still thriving as a very attractive research domain with many challenges that remain to be solved.

We would like to take this opportunity also to thank all the workshop organizers and all the referees and members of the Program Committees for their hard work in organizing the workshops and ensuring their high scientific quality.

June 2012

Marko Bajec
Johann Eder

Conference Organization

Steering Committee

Barbara Pernici	Politecnico di Milano, Italy
Oscar Pastor	Technical University of Valencia, Spain
John Krogstie	Norwegian University of Science and Technology, Norway

Advisory Committee

Arne Solvberg	Norwegian University of Science and Technology, Norway
Janis Bubenko Jr.	Royal Institute of Technology, Sweden
Colette Rolland	University of Paris 1 Sorbonne, France

General Chair

Sjaak Brinkkemper	Utrecht University, The Netherlands

Program Chairs

Jolita Ralyté	University of Geneva, Switzerland
Xavier Franch	Universitat Politècnica de Catalunya, Spain

Organizing Chair

Stanislaw Wrycza	University of Gdansk, Poland

Workshop Chairs

Marko Bajec	University of Ljubljana, Slovenia
Johann Eder	Alpen Adria Universität Klagenfurt, Austria

Workshop on Information Systems Security Engineering
WISSE

Information systems security problems are currently a widespread and growing concern that affects most areas of society, such as business, domestic, financial, government, healthcare, and so on. The scientific community has realized the importance of aligning information systems engineering and security engineering in order to develop more secure information systems. Nevertheless, there is lack of an appropriate event that promotes information systems security within the context of information systems engineering. The International Workshop on Information Systems Security Engineering (WISSE) series aims to fulfil this gap. WISSE 2012 was the second version of the workshop, with the first one being introduced with great success at the 23rd International Conference on Advanced Information Systems Engineering (CAiSE 2011). The workshop 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 relevant industrial applications, recurring challenges, problems and industrial-led solutions in the area of secure information systems engineering.

This second edition of the workshop, held in Gdańsk (Poland) on June 25, 2012, was organized in conjunction with the 24th International Conference on Advanced Information Systems Engineering (CAiSE 2012). In order to ensure a high-quality workshop, following an extensive review process, four submissions were accepted as full papers and one as a short paper addressing a large variety of issues related to secure information systems engineering.

We wish to thank all the contributors to WISSE 2012, 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 2012 Workshop Chairs for their helpful support in preparing the workshop. Finally, we thank our colleagues from the Steering Committee, Nora Cuppens, Jan Jürjens and Luis Enrique Sánchez, for initiating the workshop and contributing to its organization.

June 2012

Nadira Lammari
David G. Rosado
Haralambos Mouratidis

WISSE Organization

General Chair

Haralambos Mouratidis	University of East London, UK

Program Chairs

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
Luis Enrique Sánchez	University of Castilla-La Mancha, Spain

Publicity Chairs

Carlos Blanco	University of Cantabria, Spain
Shareeful Islam	University of East London, UK
Jérémy Milhau	Université of Paris Est, France

Program Committee

Jacky Akoka	Conservatoire National des Arts et Métiers, France
Benoit Baudry	IRISA Rennes, France
Matt Bishop	University of California, USA
Carlos Blanco	University of Cantabria, Spain
Isabelle Comyn-Wattiau	Conservatoire National des Arts et Métiers, France

Frédéric Cuppens	Telecom-Bretagne, France
Nora Cuppens	Telecom-Bretagne, France
Ernesto Damiani	Università degli Studi di Milano, Italy
Sabrina De Capitani di Vimercati	Università degli Studi di Milano, Italy
Csilla Farkas	University of South Carolina, USA
Eduardo B. Fernández	Florida Atlantic University, USA
Eduardo Fernández-Medina	University of Castilla-La Mancha, Spain
Marc Frappier	University of Sherbrooke, Canada
Steven Furnell	Plymouth University, UK
Alban Gabillon	Université de Polynésie Française, French Polynesia
Paolo Giorgini	University of Trento, Italy
Shareeful Islam	University of East London, UK
Jan Jürjens	Technical University of Dortmund, Germany
Christos Kalloniatis	University of the Aegean, Greece
Régine Laleau	University of Paris-Est, France
Nadira Lammari	Conservatoire National des Arts et Métiers, France
Yves Ledru	University of Grenoble, France
Jaejoon Lee	Lancaster University, UK
Javier López	University of Málaga, Spain
Antonio Maña	University of Málaga, Spain
Daniel Mellado	Spanish Tax Agency, Spain
Haris Mouratidis	University of East London, UK
Brajendra Panda	University of Arkansas, USA
Günther Pernul	University of Regensburg, Germany
Indrakshi Ray	Colorado State University, USA
David G. Rosado	University of Castilla-La Mancha, Spain
Guttorm Sindre	NUST, Norway
Kouichi Sakurai	Kyushu University, Japan
Luis Enrique Sánchez	University of Castilla-La Mancha, Spain

Auxiliary Reviewers

Francisco Moyano	University of Málaga, Spain
Kenichi Takahashi	Institute of Systems and Information Technologies, Japan

Workshop on Ontology, Models, Conceptualization and Epistemology in Social, Artificial and Natural Systems ONTOSE

Following the aims and scope of the ONTOSE Workshop Series to bring together researchers from different disciplines, ONTOSE 2012 covered a wide range of topics related to ontologies and their usage in different kinds of software systems. The sixth edition of the workshop was hosted for the fourth year as a satellite event of the CAISE Conference.

The Program Committee accepted 5 long and 3 short papers out of 16 submissions.

The three long papers are focused on economy: the first one proposes an ontology-based, event-driven architecture for integrating information, processes and services to coordinate delivery; the second paper describes how topic maps can be used to build ontologies of economic indicators useful in the analysis of economic ratios on account and the semantic connections between them; the third one presents an ontology of contracts which introduces some important innovations having positive impacts on enterprises. The fourth long paper is about a temporal model capturing important concepts involved in the representation of time-aware activities and their relations. The model is described by means of UML formalization, enriched with OCL constraints where needed. Finally, in the fifth long paper an enhanced version of the MoKi tool equipped with an ontology evolution approach that permits one to evolve an ontology by providing a mechanism for facing the tracking challenge is presented.

The short papers are about ongoing research project or case studies involving ontology: one of them describes an ontology of events that, combined with an already existing ontology of crowds, should enable one to build application scenarios useful in the analysis of people participating in big events and how they group; the second paper concerns the important topic of ontology enrichment and alignment, and proposes a method for discovering new concepts and examining the equivalence of properties in different LOD description schemas by using open information extraction techniques on Web resources. The last short paper focuses on the modeling of scientific information, comparing two of the more widespread models, and provides directions for mapping them in a way that enables clients to integrate data coming from heterogeneous sources.

We wish to thank all the people who helped make ONTOSE 2012 a success. Namely, we thank all the authors who submitted their valuable contributions and the CAiSE workshop organizers (Marko Bajec and Johann Eder) who actively supported us during the entire process.

June 2012

Fabio Sartori
Miguel-Angel Sicilia
Christian Kop

ONTOSE Organization

Program Committee

Program Chairs

Fabio Sartori	University of Milano-Bicocca, Italy
Miguel-Angel Sicilia	University of Alcalá, Spain
Christian Kop	Alpen Adria University Klangenfurt, Austria

Organizing Chair

María-Cruz Valiente	University of Alcalá, Spain

Referees

S. Agarwal (Germany)
J. A. Olivas (Spain)
S. Bandini (Italy)
D. Benslimane (France)
J. Bermejo-Alonso (Spain)
Pierre Bourque (Canada)
Pierre-Jean Charrel (France)
J. Ebert (Germany)
W. Esswein (Germany)
J. Eder (Austria)
S. España Cubillo (Spain)
R. Feldmann (USA)
A. Felfernig (Austria)
F. Ferrarotti (New Zealand)
A. Finkelstein (UK)
G. Friedrich (Austria)
F. Gargouri (Tunisie)
A. Ghose (Australia)
A. Ginige (Australia)
O. Gotel (USA)
P. Haase (Germany)
W. Hesse (Germany)
D. Jannach (Germany)
P. Johannesson (Sweden)
R. Kaschek (Faroe Islands)
M. Lytras (Greece)
N. Manouselis (Greece)
H. C. Mayr (Austria)
J.J. Charles Meyer (The Netherlands)
D. Micucci (Italy)
R. Mittermeir (Austria)
C. Pahl (Ireland)
J. Palomaki (Finland)
F. Silva Parreiras (Germany)
D. Rodrigues (Spain)
F. Ruiz Gonzalez (Spain)
C. Scharff (USA)
C. Stracke (Germany)
M. Stumptner (Australia)
F. Tisato (Italy)
M. Zanker (Austria)

10th International Workshop on System/Software Architectures IWSSA

As the 10^{th} in the series of the International Workshop on System/Software Architectures (IWSSA 2012), this workshop reflected the importance of systems and software architectures as a touchstone for many system and software engineering (SSE) projects. In particular, this workshop addressed, and bridged the gap between, system engineering and software engineering, through seamless integrations of system architectures and software architectures, which by and large have been developed and used disparately. The increasing connectivity between systems, hence their complexity, and the successive emergence of new technologies and computation paradigms, and the advent of new development approaches lend new opportunities for academicians, practitioners and even politicians to steadily strive to come up with new system designs or even revisit old ones.

In this context, service-oriented and/or service-driven architectures are gaining momentum within the SSE community and even changing the way in which system operations are conceived. Indeed, "services thinking" almost equates to not developing any more, but orchestrating and choreographing the functionalities that have been developed in the past, typically by means of well-defined interfaces. Interoperability, heterogeneity, etc., are commonplace in this context. IWSSA workshops especially focus on the enactment/connection of the different kinds of system and software requirements into/to concrete design issues for system/software architectures, with a special emphasis on non-functional requirements analysis.

This was the second time IWSSA was held in association with the International Conference on Advanced Information Systems Engineering (CAiSE 2012). The present edition received a total of 20 paper submissions out of which seven papers were finally selected as full papers and three as short papers. This fact does not belittle the quality of the rest of the papers which were also well rated and we hope to be able to receive submissions of improved versions to subsequent editions. We would like to sincerely thank the CAiSE Workshops Chairs, Johann Eder and Marko Bajec, for their support, the IWSSA 2012 Program Committee members and external reviewers for their excellent work during the review process, and most importantly the authors of the papers for their very interesting and high-quality contributions, which makes it possible this workshop series every year.

June 2012

Lawrence Chung
Manuel Noguera
Nary Subramanian
José Luis Garrido

IWSSA Organization

Executive Committee

Workshop Chairs

Lawrence Chung	University of Texas at Dallas, USA
Manuel Noguera	University of Granada, Spain
Nary Subramanian	University of Texas at Tyler, USA
José Luis Garrido	University of Granada, Spain

Organizing Chair

Kawtar Benghazi	University of Granada, Spain

Program Committee

W. van der Aalst, The Netherlands
Philippe Aniorte, France
Hernán Astudillo, Chile
Doo-Hwan Bae, Korea
Joseph Barjis, The Netherlands
Jaelson Castro, Brazil
Roger Champagne, Canada
François Coallier, Canada
Kendra Cooper, USA
Rafael Corchuelo, Spain
Lirong Dai, USA
Sergiu Dascalu, USA
Yannis A. Dimitriadis, Spain
Jing Dong, USA
Jesús Favela, Mexico
Juan Fernández-Ramil, UK
Rubén Fuentes, Spain
Lars Grunske, Australia
Fred Harris, USA
María V. Hurtado, Spain
Stan Jarzabek, Singapore
Li Jiang, Australia
Carlos Juiz, Spain
Pericles Loucopoulos, UK
María D. Lozano, Spain
Chung-Horng Lung, Canada
Stephen J. Mellor, USA
Tommi Mikkonen, Finland
Ivan Mistrík, Germany
Masaki Murakami, Japan
Selmin Nurcan, France
Sergio F. Ochoa, Chile
Patricia Paderewski, Spain
Sooyong Park, Korea
Oscar Pastor, Spain
Fabio Paternò, Italy
Claudia Raibulet, Italy
María L. Rodríguez, Spain
Gustavo Rossi, Argentina
Vespe Savikko, China
Michael Shin, USA
Yeong Tae Song, USA
Sam Supakkul, USA
Dalila Tamzalit, France
Sebastián Uchitel, Argentina
Andrea Zisman, UK

External Reviewers

Suntae Kim, Korea
Isaac Lera, Spain
João Pimentel, Brazil
Ed Barkmeyer, USA
Carlos Guerrero, Spain

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

We are pleased to present the papers of the First Workshop on Human-Centric Process-Aware Information Systems (HC-PAIS 2012) held in conjunction with the 24th International Conference on Advanced Information Systems Engineering (CAiSE 2012). The objective of the HC-PAIS workshop was to extend the spectrum of the main conference by giving researchers and practitioners a forum to present and discuss cutting-edge research in the field of information systems engineering. The HC-PAIS workshop was themed on the human-centric view on information systems and the interplay of technological and work psychological aspects in the context of process-aware information systems (PAIS). It aimed to offer a portal for researchers, discussions, and contributions of interdisciplinary research dedicated to the investigation of particularly human-centric processes that demand human judgment, skills, competencies, experiences, and discernment.

This edition of the workshop attracted papers from six different countries including Australia, Austria, Germany, Italy, Poland, and Spain. We received nine papers, from which the top three of the best rated papers were selected, an acceptance rate of 33%. The accepted papers addressed various aspects such as gesture-based process modeling, creativity in process design, and user-centric abstraction of workflow logic.

In "Towards Gesture-Based Process Modeling on Multi-Touch Devices," Jens Kolb, Benjamin Rudner and Manfred Reichert present experimental research results of multi-touch process modeling and suggest a core gesture set for modeling and adapting processes.

Kathrin Figl and Barbara Weber investigate in their paper "Individual Creativity in Designing Business Processes" how process model competence and individual creativity style as well as creative capacity influence creativity in a business process redesign task by means of an empirical study.

In "User-Centric Abstraction of Workflow Logic Applied to Software Engineering Processes," written by Gregor Grambow, Roy Oberhauser and Manfred Reichert, an approach to model user decisions influencing the actual trace of workflows is presented and applied to a concrete application scenario.

We thank the authors for submitting their papers, the Program Committee members for reviewing the papers, and the CAiSE 2012 Workshop Co-chairs and the Organizing Committee for all their support.

June 2012

Sonja Kabicher-Fuchs
Jan Recker
Stefanie Rinderle-Ma

HC-PAIS Organization

HC-PAIS 2012 was a collaboratively organized workshop of the Research Group Workflow Systems and Technology, University of Vienna, and the Business Process Management Group, Queensland University of Technology.

Workshop Chairs

Sonja Kabicher-Fuchs	University of Vienna, Austria
Jan Recker	Queensland University of Technology, Australia
Stefanie Rinderle-Ma	University of Vienna, Austria

Program Committee

Schahram Dustdar	Vienna University of Technology, Austria
Kathrin Figl	Vienna University of Economics, Austria
Simone Kriglstein	University of Vienna, Austria
Jan Mendling	Vienna University of Economics, Austria
Manfred Reichert	Ulm University, Germany
Hajo A. Reijers	Eindhoven University of Technology, The Netherlands
Stefan Seidel	University of Lichtenstein, Lichtenstein
Kenia Sousa	Intrasoft International, Belgium
Joyce Nakatumba	Eindhoven University of Technology, The Netherlands
Barbara Weber	University of Innsbruck, Austria
Moe Wynn	Queensland University of Technology, Australia

Workshop on Governance, Risk and Compliance in Information Systems GRCIS

The importance of governance and associated issues of compliance and risk management is well recognized in enterprise systems. This importance has dramatically increased over the last few years as a result of numerous events that led to some of the largest scandals in corporate history. The governance, risk and compliance market is estimated to be worth over US$32 billion. Tool support for governance, risk and compliance-related initiatives is provided by over 100 software vendors; however, while the tools have on average tripled in price since 2003, they are often insufficient to meet organizational needs. At the same time, there is an increasing complexity in the facilitation of compliant business processes, which stems from an increasing number of regulations, frequent and dynamic changes, as well as shared processes and services executing in highly decentralized environments.

In the age of outsourcing, dynamic business networks, and global commerce, it is inevitable that organizations will need to develop methods, tools and techniques to design, engineer and assess processes and services that meet regulatory, standard and contractual obligations. Governance, risk and compliance (GRC) can be expected to play a significant part in several applications. This area is emerging as a critical and challenging field of research and innovation. It introduces, among others, the need for new or adapted modeling approaches for compliance requirements, extension of process and service modeling and execution frameworks for compliance and risk management, and detection of policy violations. In addition, it introduces questions relating specifically to the use of technology to support compliance management. For example, how auditors and regulators can put into use techniques such as continuous monitoring and data analysis to assess whether an organization complies with relevant rules and regulations, or how technology can be used to support assessment of design and operational effectiveness of controls.

The GRCIS workshop aims to provide a forum for researchers from diverse areas related to GRCIS to make a consolidated contribution in the form of new and extended methods that address current challenges. Three full research papers were accepted for publication this year, representing a 30% acceptance rate.

June 2012

Marta Indulska
Shazia Sadiq
Michael zur Muehlen
Yao-Hua Tan

GRCIS Organization

Workshop Chairs

Marta Indulska	The University of Queensland, Australia
Shazia Sadiq	The University of Queensland, Australia
Michael zur Muehlen	Stevens Institute of Technology, USA
Yao-Hua Tan	Delft University of Technology, The Netherlands

Associate Editor

Sietse Overbeek	Delft University of Technology, The Netherlands

Program Committee

Ahmed Awad	Cairo University, Egypt
Sami Bhiri	National University of Ireland, Ireland
Dickson K.W. Chiu	Dickson Computer Systems, Hong Kong
Anne Cleven	University of St.Gallen, Switzerland
Francien Dechesne	Delft University of Technology, The Netherlands
Philip Elsas	ComputationalAuditing.com, Canada
Ralph Foorthuis	Utrecht University, The Netherlands
G.R. Gangadharan	Telematica Institute, The Netherlands
Anke Gericke	University of St. Gallen, Switzerland
Aditya Ghose	University of Wollongong, Australia
Guido Governatori	NICTA, Australia
Joris Hulstijn	Delft University of Technology, Thauris B.V., The Netherlands
Thao Ly	University of Ulm, Germany
Olivier Perrin	Nancy 2 University, France
Antonio Ruiz-Cortes	University of Seville, Spain
Michael Schermann	Technical University Munich, Germany
Leon Van Der Torre	ILIAS, Luxembourg
Hans Weigand	Tilburg University, The Netherlands
Roel Wieringa	University of Twente, The Netherlands

Workshop on Enterprise and Organizational Modeling and Simulation EOMAS

Enterprises and other organizations of the twenty-first century are crucial components in delivering services to society and contributing to the economic prosperity and quality of life for people. Service is delivered when an enterprise or organization is conducting its business within its business environment using optimal technology and knowledge. With the growing complexity of modern business processes and continuously changing business environments, business and software engineering requires advanced approaches with properties such as ability for reengineering, scalability, adaptability, and reimplementation.

However, an extended enterprise and organizational study involves both analysis and design activities, in which modeling and simulation play a prominent role. The growing role of modeling and simulation is attracting the serious attention of researchers. Modeling and simulation are tools and methods that are effective, efficient, economical, and widely used in business and software engineering, organizational studies, knowledge transfer and business process management. Complementary insights of modeling and simulation constitute a whole cycle of elaboration of these complex sociotechnical systems.

The EOMAS Workshop (Enterprise Organization Modeling and Simulation) was founded in 2005 by SIGMAS (Special Interest Group on Modeling and Simulation), which is a formal entity of the AIS (Association for Information Systems). EOMAS participants are an international community of professionals from a broad spectrum of disciplines who want to share the benefits of their knowledge and expertise. The main objective of the annual EOMAS workshops is to fill in the gap between business and IS and to minimize the failure rate of information systems through the application of proper modeling and simulation before the system is built. EOMAS aims to benefit and serve the community and advance the discipline of modeling and simulation in the broad context of computing (e.g., software engineering, information systems and information technology) and contribute to the theory of modeling and simulation, the development of methodologies, tools and techniques. A collection of the fully reviewed papers presented at the 8th EOMAS workshop were selected for publication in these proceedings.

June 2012

Vojtech Merunka
Martin Molhanec

EOMAS Organization

The EOMAS workshop is organized annually as an international forum for researches and practitioners in the field of enterprise and organizational modeling and simulation. Organization of this workshop and peer reviews of the contributions made to this workshop were accomplished by an international team of researchers in the field of enterprise modeling and simulation, business and informatics.

Workshop Organizers

Chair

Vojtech Merunka	Czech University of Life Sciences Prague, CTU in Prague, Czech Republic

Co-chairs

Anna Bobkowska	Gdansk University of Technology, Poland
Martin Molhanec	Czech Technical University in Prague, Czech Republic
Srini Ramaswamy	ABB Corporate Research, India

Program Committee

Zacharoula Andreopoulou	Aristotle University of Thessaloniki, Greece
Antonia Albani	University of St. Gallen, Switzerland
Jean-Paul Arnaout	Lebanese American University, Lebanon
Joseph Barjis	Delft University of Technology, The Netherlands
Ygal Bendavid	Polytechnique Montreal, Canada
Kawtar Benghazi	University of Granada, Spain
Ilona Bluemke	Warsaw University of Technology, Poland
Peter Bollen	Maastricht University, The Netherlands
Mahmoud Boufaida	Mentouri University of Constantine, Algeria
Tatiana Bouzdine-Chameeva	Bordeaux Management School, France
Zeynel Cebeci	Cukurova University, Turkey
Rodney Clarke	University of Wollongong, Australia
Jan Dietz	Delft University of Technology, The Netherlands
Tillal Eldabi	Brunel University, UK
Samuel Fosso Wamba	University of Wollongong, Australia
José Luis Garrido Bullejos	University of Granada, Spain
Ivan Georgiev	Trakia University, Bulgaria

Rafael Gonzalez	Javeriana University, Colombia
Ashish Gupta	Minnesota State University Moorhead, USA
Bogumila Hnatkowska	Wroclaw University of Technology, Poland
P. Radha Krishna	Infosys Technologies Ltd., India
Emil Krsak	University of Zilina, Slovakia
Peggy Daniels Lee	Indiana University-Kelley School of Business, USA
Prabhat Mahanti	University of New Brunswick, Canada
Karol Matiasko	University of Zilina, Slovakia
Alta van der Merwe	University of South Africa, South Africa
Jaroslava Mikulecka	University of Hradec Kralove, Czech Republic
Navonil Mustafee	Swansea University, UK
Manuel Noguera	University of Granada, Spain
Robert Pergl	Czech University of Life Sciences in Prague, Czech Republic
Ghaith Rabadi	Old Dominion University, USA
Gregory W. Ramsey	Morgan State University, USA
Han Reichgelt	Southern Polytechnic State University, USA
Peter Rittgen	University College of Boras, Sweden
Victor Romanov	Plekhanov University of Economics, Russia
Irina Rychkova	University Paris 1 Pantheon – Sorbonne, France
Michail Salampasis	ATEI of Thessaloniki, Greece, University of Vienna, Austria
Mamadou Seck	Delft University of Technology, The Netherlands
Natalia Sidorova	Eindhoven University, The Netherlands
Antonin Slaby	University of Hradec Kralove, Czech Republic
Michel Soares	Federal University of Uberlandia, Brazil
David Sundaram	University of Auckland, New Zealand
Yutaka Takahashi	Senshu University, Japan
Jose Tribolet	Technical University of Lisbon, Portugal
Georgios Tegos	ATEI of Thessaloniki, Greece
Milan Turcani	Constantine the Philosopher University in Nitra, Slovakia
Stergios Tzortzios	University of Thessaly, Greece

Sponsoring Institutions

- SIGMAS (Special Interest Group on Modeling and Simulation of the Association for Information Systems) – in collaboration
- CAiSE 2012 (International Conference on Advanced Information System Engineering)
- TU Delft (Delft University of Technology, Department of System Engineering)

7th International Workshop on Businness/IT Alignment and Interoperability BUSITAL

Nowadays, information services are core assets for companies having a sustainable competitive advantage. The continuous growth of information and communication technologies-enabled innovations provides organizations with new efficient mechanisms for communication, information sharing, resource management and planning, and helps them to explore new market opportunities. The special theme of the 24th edition of CAiSE was "Information Services" where the notion of service plays a more and more extensive role in the enterprise development. However, we feel that 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, 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) an organization on behalf of a customer; and in computer science a service is a programmable, self-describing, encapsulated, and loosely coupled functions accessed and invoked over the Internet.

As a consequence, the governance of information services in this increasing evolving scenario asks for new models of alignment not only with 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, the recent decade brought yet another wave of ICT innovations: cloud, smart and mobile technologies are rapidly integrating 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 seventh edition of the BUSITAL workshop clearly demonstrated the increasing interest of business-IT alignment in the management community.

June 2012

Birger Andersson
Christian Huemer
Irina Rychcova
Gianluigi Viscusi

BUSITAL Organization

Organizing Chairs

Birger Andersson	Stockholm University, Sweden
Christian Huemer	Vienna University of Technology, Austria
Irina Rychcova	Centre de Recherches en Informatique, Paris 1, France
Gianluigi Viscusi	University of Milano-Bicocca, Italy

Program Committee

Joseph Barjis, The Netherlands
Judith Barrios Albornoz, Venezuela
Alessio Maria Braccini, Italy
Eric Dubois, Luxembourg
Pascal van Eck, The Netherlands
Ananda Edirisuriya, Sri Lanka
Jaap Gordijn, The Netherlands
Tharaka Ilayperuma, Sri Lanka
Paul Johannesson, Sweden
John Krogstie, Norway
Kalle Lyytinen, USA
Raimundas Matulevicius, Latvia
Selmin Nurcan, France
Andreas L. Opdahl, Norway
Michael Petit, Belgium
Yves Pigneur, Switzerland
Jolita Ralyte, Switzerland
Guttorm Sindre, Norway
Janis Stirna, Sweden
Alain Wegmann, Switzerland
Hans Weigand, The Netherlands
Roel Wieringa, The Netherlands
Eric Yu, Canada
Jelena Zdravkovic, Sweden

Referee

M. Comerio

International Workshop on Agility of Enterprise Systems (AgilES 2012)

An enterprise system provides a technical platform that enables organizations to run their business more effectively. As strategic and operating conditions become increasingly turbulent due to factors such as hyper-competition, increasing demands from customers, regulatory changes, and technological advancements, the ability to sense relevant change and respond appropriately becomes an important determinant of enterprise success. The agility of an enterprise system typically refers to the ability of an organization and the information system supporting its business processes to rapidly adapt to market and environmental changes in productive and cost-effective ways.

The Workshop AgilES was started with this purpose in mind: to bring the results of rigorous scientific research and experience of industrial practices on enterprise systems, especially about enhancing agility of enterprise systems, to researchers and practitioners, so that they can create a shared vision and bridge any communication and understanding gaps. The call for papers attracted high-quality submissions from North America and Europe. Considering that the concept of the agility of enterprise systems is still in its infancy stage, the Program Committee decided to accept two papers and reserved more time during the workshop for discussion.

AgilES 2012 was a team effort. We would like to thank the Program Committee and external reviewers who worked extremely hard to get things done in time. The CAiSE conference organization teams were also very instrumental in handling the co-hosted workshops and deserve special thanks. Special thanks also go to Marko Bajec, Johann Eder and Vojtech Merunka for their support. Last, but not least, we thank all authors and participants for contributing to a rich experience.

We hope that you will find the workshop proceedings interesting. We also hope that in the years to come, AgilES will become a platform for dialogue on new concepts and research on enterprise systems and their agility.

June 2012 Xiaocheng Ge

AgilES Organization

Organizers

Xiaocheng Ge	University of York, UK
Xiaofeng Wang	Free University of Bozen, Italy
Nour Ali	Lero, Irish Software Engineering Research Centre, Ireland

Program Committee

Rami Bahsoon	University of Birmingham, UK
Juan Garbajosa	Technical University of Madrid (UPM), Spain
Ivan Mistrik	Independent Consultant, Germany
Ricardo Martinho	Polytechnic Institute of Leiria, Portugal
Richard Paige	University of York, UK
Raman Ramsin	Sharif University of Technology, Iran
Juha Rikkilae	Free University of Bozen, Italy
Dina Salah	University of York, UK
Carlos Solis	FEXCO, Ireland
Masoumeh Taromirad	University of York, UK

Table of Contents

WISSE: Workshop on Information Systems Security Engineering

ONTOSE: Workshop on Ontology, Models, Conceptualization and Epistemology in Social, Artificial and Natural Systems

ONTOSE Session 1: Ontologies Based on Time and Events

ONTOSE Session 2: Ontology Evolution, Mapping and Enrichment

ONTOSE Session 3: Ontologies in Economics and Trade

IWSSA: Workshop on System/Software Architectures

IWSSA Session 1

IWSSA Session 2

IWSSA Session 3

IWSSA Session 4

GRCIS: Workshop on Governance, Risk and Compliance in Information Systems

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

EOMAS: Workshop on Enterprise and Organizational Modeling And Simulation

EOMAS Session 1

EOMAS Session 2

EOMAS Session 3

EOMAS Session 4: Short Papers

BUSITAL: Workshop on BUSinness/IT ALignment and Interoperability

BUSITAL Session 1: Alignment in IT Services

BUSITAL Session 2: Alignment Engineering

BUSITAL Session 3: Alignment Maturity Models

BUSITAL Session 4: Short Papers

AgilES: Workshop on Agility of Enterprise Systems

Towards Definition of Secure Business Processes

Olga Altuhhova, Raimundas Matulevičius, and Naved Ahmed

Institute of Computer Science, University of Tartu
J. Liivi 2, 50409 Tartu, Estonia
olgaaltuhhova@hotmail.com, {rma,naved}@ut.ee

Abstract. Business process modelling is one of the major aspects in the modern system development. Recently business process model and notation (BPMN) has become a standard technique to support this activity. Although BPMN is a good approach to understand business processes, there is a limited work to understand how it could deal with business security and security risk management. This is a problem, since both business processes and security concerns should be understood in parallel to support a development of the secure systems. In this paper we analyse BPMN with respect to the domain model of the IS security risk management (ISSRM). We apply a structured approach to understand key aspects of BPMN and how modeller could express secure assets, risks and risk treatment using BPMN. We align the main BPMN constructs with the key concepts of the ISSRM domain model. We show applicability of our approach on a running example related to the Internet store. Our proposal would allow system analysts to understand how to develop security requirements to secure important assets defined through business processes. In addition we open a possibility for the business and security model interoperability and the model transformation between several modelling approaches (if these both are aligned to the ISSRM domain model).

Keywords: Business process model and notation (BPMN), Security risk management, Alignment of modelling languages, Information systems.

1 Introduction

Business process modelling takes an important part when developing information systems (IS). It helps specify standard and optimised workflows of organisation. The business processes that involve many participants, their communications, necessary resources and their usage not only extend organisational competiveness but also increase business vulnerabilities. Thus, understanding and modelling of IS security becomes an important activity during IS development. Security refers to the capability of a product, i.e., IS, to protect data and information against the unauthorised access by persons or systems that have intention to harm it.

Identification of the security requirements is typically performed only after the business process has been defined. Furthermore, it is observed [12] that security considerations often arise most usually during *implementation* or *maintenance* stages. Firstly, this means that security engineers get little feedback about the need for system security. Secondly, security risks are very hard to calculate: security-critical systems

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 1–15, 2012.

are characterised by the fact that the occurrence of a successful attack at one point in time on a given system increases the likelihood that the attack will be launched subsequently at another system point. This is a serious hindrance to secure system development, since the early consideration of security (e.g., when defining the business processes) allows engineers to envisage threats, their consequences and design countermeasures. Then the system design and architecture alternatives, that do not offer a sufficient security level, could be discarded.

Although there exists few attempts to introduce notations to address security at the business process modelling (i.e., [16] [19] [20]) or to relate business process and security requirements modelling (i.e., [17]), these are rather at the coarse-grained level. In principle, the approaches do not illustrate guidelines on how to advance from one security aspect to another, or how to understand security concerns and define security requirements.

In this work we consider Business Process Model and Notation (BPMN, version 2.0) [18] [21], a multi-vendor standard controlled by the Object Management Group [24]. The primary purpose of BPMN is modelling of the business processes. Like in other modelling languages, BPMN notations are linked to a semantic model, which means that each shape has a specific meaning, and defined rules to connect objects. In this work our goal is *not* to develop new modelling approach for security, but rather to understand (*i*) how business activities expressed using BPMN could be annotated with the security concerns; (*ii*) how BPMN could be used to define security requirements; and (*iii*) how the BPMN language itself could be used to reason for the security requirements through illustration of the potential security risks. In this paper we specifically address the second (*ii*) and third (*iii*) aspect.

To achieve our goal we have selected a domain model [7] [15] for IS security risk management (ISSRM) and have aligned the BPMN constructs to the concepts of this domain model. We result in a grounded and fine-grained reasoning for extensions of BPMN toward secure business processes. In addition we present our analysis through an illustrative example; thus, in this way we end up with guidelines for the BPMN application to analyse security risks.

The paper structure is as follows: in Section 2 we give the background to our study. Based on the running example in Section 3 we investigate BPMN following the ISSRM process. In Section 4 we present an alignment of BPMN constructs to the concepts of ISSRM. In Section 5 we discuss our finding, related work and conclude our study.

2 Background

2.1 Security Analysis Methods

To model secure systems, different security risk management approaches are developed. For instance, CORAS is a model-driven approach [4], which includes a systematic guidance for security risk analysis. The Tropos Goal-Risk framework [2] supports modelling, assessing and treating risks on the basis of the likelihood and severity of failures. This framework consists of three conceptual layers – strategy, event, and treatment to assess the risk of some events and evaluate the effectiveness of treatments. CoBRA [23] provides tools for quantitative risk evaluation and consulting. Using CoBRA developers reduce the losses that might result from security problems.

Risk-based requirements elicitation and prioritization (RiskREP) [10] is an iterative process for managing IT security risks. It combines the results of requirements analysis and risk analysis. The analysis is carried on in four steps: elicitation of quality goals, security risk analysis, countermeasure definition, and prioritisation.

In this work we situate our analysis at the *fine-grained* level in order to outline the capabilities of BPMN to deal with security. Our goals are to explore how we could apply BPMN to model security when considering business process modelling, and to suggest some potential BPMN extensions towards security. To ground our analysis we select the *domain model* for *information systems security risk management* (ISSRM) [7] [15]. The ISSRM approach also includes process guidelines that help identify the vulnerable assets, determine their security objectives, assess the risks, and elicit security requirements to mitigate these risks.

2.2 ISSRM Domain Model

Since the ISSRM domain model [7] [15] (shown in Fig. 1) is an important artefact to analyse BPMN in this paper, we will briefly introduce its major concepts.

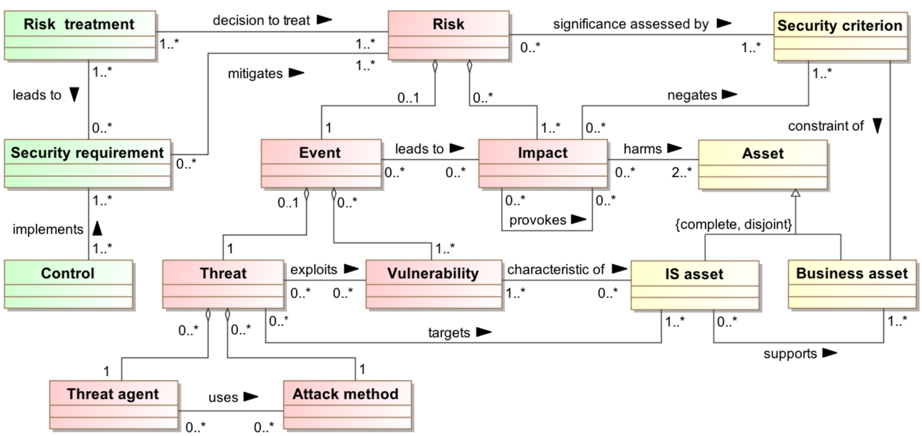


Fig. 1. The ISSRM Domain Model (adapted from [7] [15])

Assets-related concepts describe organisation's assets and their security criteria. Here, an *asset* is anything that is valuable and plays a vital role to accomplish organisation's objectives. A *business asset* describes the information, processes, capabilities and skills essential to the business and its core mission. An *IS asset* is the IS component, valuable to the organisation since it supports business assets. A *security criterion* is the property or constraint on business assets describing their security needs, which are, typically, expressed through *confidentiality*, *integrity* and *availability.*

Risk-related concepts introduce a risk definition. A *risk* is composed of a threat with one or more vulnerabilities that leads to a negative impact on one or more assets by harming them. An *impact* is the consequences of an event that negates the security criterion defined for business assets in order to harm assets. An *event* is an aggregation of threat and one or more vulnerabilities. A *vulnerability* is the characteristics of IS assets that expose weakness or flaw. A *threat* is an incident initiated by a threat agent using attack method to target one or more IS assets by exploiting their vulnerabilities. A *threat agent* is an agent who has means to harm intentionally IS assets. An *attack method* is a standard means by which a threat agent executes threat.

Risk-treatment related concepts describe the concepts to treat risk. A *risk treatment* is a decision (e.g., *avoidance*, *reduction*, *retention*, or *transfer*) to treat the identified risk. A *security requirement* is the refinement of a risk treatment decision to mitigate the risks. A *control* designates a means to improve the security by implementing the security requirements.

Application guidelines. The ISSRM application follows the general risk management process. It is based on the existing security standards, like [3] [6] [11] [22]. It is an iterative process consisting six steps. Firstly, a developer needs to *define the organisational context and assets* that needs to be secured. Then, one *determines security objectives* (e.g., confidentiality, integrity, and availability) based on the level of protection required for the identified assets. Next, *risk analysis and assessment* help identify potential risks and their impacts. Once risk assessment is performed *risk treatment decision* should be taken. This would result in *security requirements definition*. Security requirements are *implemented* into *security controls*. The risk management process is iterative, because new security controls might open the possibility for new (not yet determined) security risks.

2.3 Research Method

The ISSRM domain model [7] [15] was developed during the *step 1* and *step 2* as illustrated in the research method in Fig. 2. The main goal of the step 1 was to identify the most important concepts of the security risk domain. The literature on the risk management standards [3] [11], security-related standards [6] [22], security risk management methods [1] [4] and software engineering frameworks [8] [9] was considered. Based on this analysis, a conceptual model (see Fig. 1) is defined. In addition each concept (i.e., class and association) is complemented with definition. In this work we focus on the third step. As discussed in [7] [15], most of the modelling languages appear to overlook security risk management (despite of few reports in [5] [13] [14]). In this paper we report on the BPMN means to address security risk management. The outcome of our analysis is the direct input for the fourth step where the modelling language could be extended with the security risk management constructs and its usage adjusted to the guidelines of the risk management process. This work is a part of the larger effort to develop a systematic model transformation-based security risk-driven method for secure system development.

2.4 BPMN

The application of BPMN modelling is divided into three model usage levels [21]. *Analytical modelling* describes the activity flow. *Executable modelling* is targeted to the system developing. In this paper our scope is *descriptive modelling*, which concentrates on business process by documenting the major business flows.

The major BPMN graphical constructs (concrete syntax) for the *descriptive modelling* are listed in Fig. 3. The extract of the BPMN abstract syntax (based on [18]) classifies BPMN graphical constructs and illustrate the relationships between them in Fig. 4 and 5.

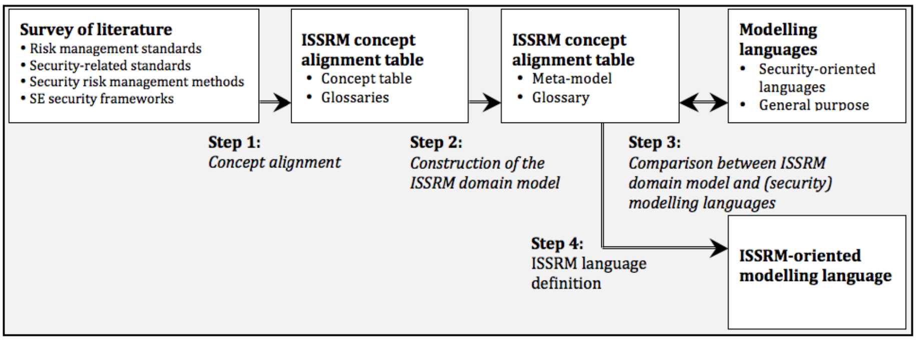


Fig. 2. A Research Method for ISSRM-oriented Modelling Languages (adapted from [15])

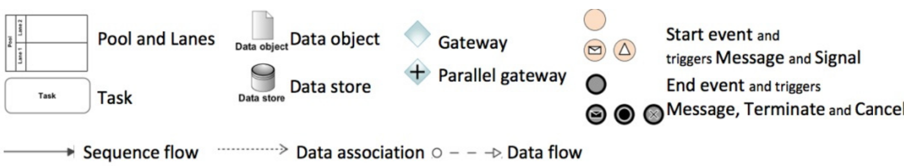


Fig. 3. BPMN Concrete Syntax (Descriptive Modelling)

BPMN includes four major categories of constructs (Fig. 4): *flow objects*, *containers*, *flows* and *artefacts*. The *flow objects* describe the atomic units of a process using *events*, *tasks* and *gateways*. An *event* indicates *start* or *end* of a process path; it can be *triggered* or *non-triggered*. A *task* is an atomic activity that has no internal sub-parts defined by the model. In some cases, the *task* can also represent the sub-process, a compound activity with sub-parts. The control of the divergence and convergence of sequence flows is realised by the *gateways*. The BPMN *containers* are *pools* and *lanes*. They both play a role of object holders. However, the *pool* shows the message flow between the process and external participants. The *lane* is a subdivision of a process used to organise flow elements belonging to different categories, and also represents a performer role or an organisational unit. The BPMN *artefacts* include such constructs as *data objects*, *data stores* and *annotations*. *Data objects* define what data is required or produced by activities. *Data stores* describe how data are stored.

Relationships (Fig. 5) between different BPMN constructs are defined using *flows*, which include *sequence flows*, *data flows*, and *data association flows*. For instance, the *sequence flows* link together the BPMN activities, gateways, and events within a single pool. The *data flows* show the input/output between pools. Finally, the *data association flows* link together the BPMN tasks and artefacts (i.e., data objects, data stores, and annotations).

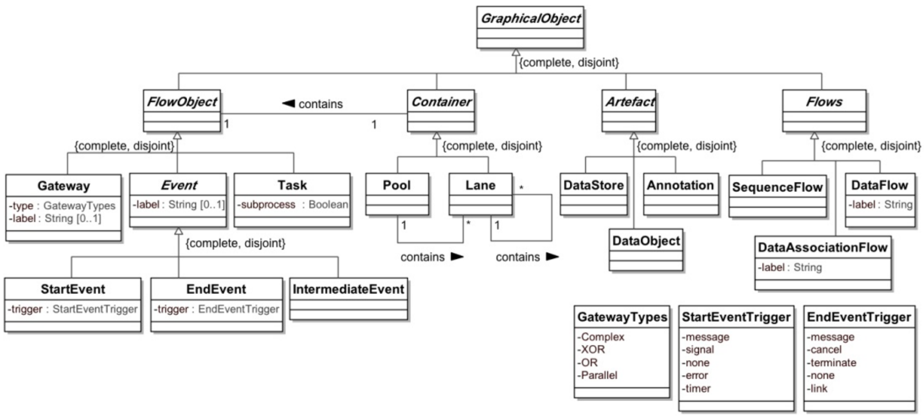


Fig. 4. The BPMN Abstract Syntax: Concept Classification

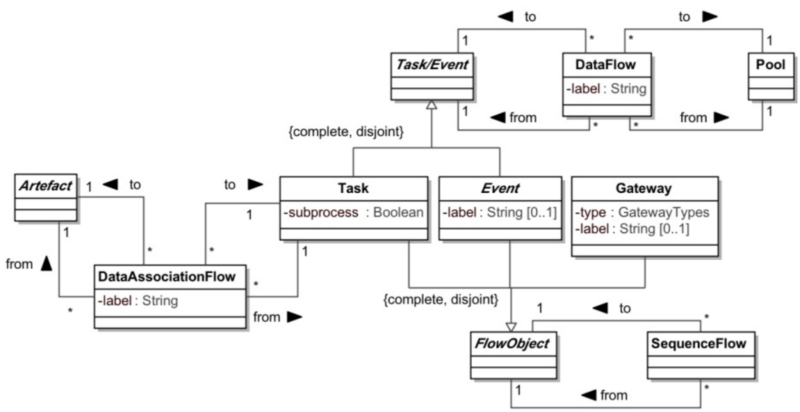


Fig. 5. The BPMN Abstract Syntax: Relationships[1]

3 Security Risk Modelling with BPMN

In this section we will follow the ISSRM process to investigate security risks in a running example modelled using BPMN. We will show which BPMN constructs could be used to address concepts of the ISSRM domain model. Our running example is an *online registration process of the Internet store*.

Context and Asset Identification. Let's consider the following situation where the potential User (*pool* User in Fig. 6) wishes to start using the Internet store system (*pool* System). In order to get registration details, user sends a message with an inquiry to the

---

[1] Here we do not define the explicit integrity constraints of the abstract syntax. But these exist, especially, to strengthen the flow relationships. For instance, the data association flow could only be defines between the artefacts and task; the data flow could only be defined between the pool and task/event, and similar.

system administrator. After the message is accepted (*task* Accept message) and read (*task* Read message) by the administrator, the guidelines (*data flow* Demand for registration) are sent (*task* Send answer) back to the user.

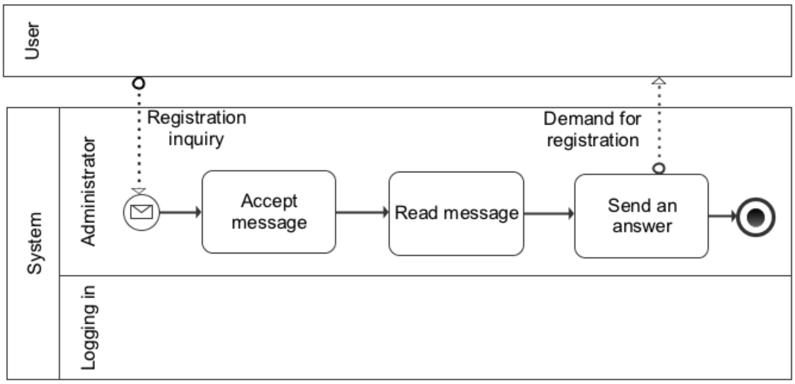


Fig. 6. Message Handling Process

In Fig. 7 we present a user registration process. After receiving the guidelines, the user registers to the Internet store by submitting his data (*data flow* User info). The system, then, accepts registration information (which includes data on the preferred Username and Password) and includes it into the database (*task* Insert data to DB).

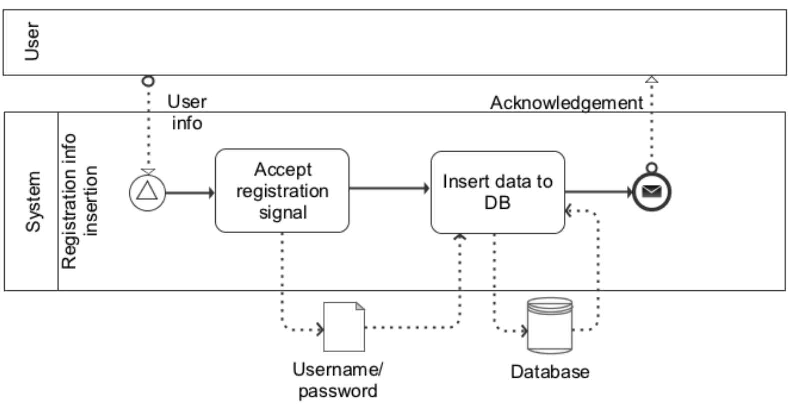


Fig. 7. User Registration Process

After registering the valid Username and Password, the user is able to login to the system. The system checks the username and the password. If these match, the user gets the *success* signal and is able to use the Internet store system. Otherwise the user gets a notification about the failure.

Determination of Security Objectives. In this scenario we can identify several major assets that needs protection against security risks. Firstly, we need to ensure *confidentiality of username and password*. If confidentiality is revealed the system violators could use the user's personal data for not intended purposes. In addition we need to ensure *integrity of all the business processes*. If integrity is broken the system might be used not according to its purpose.

Risk Analysis and Assessment. In Fig. 8 we model a potential security risk scenario. Let's say, that there exists a violator (presented as the BPMN *pool* Violator) who would like to login to the system without registering his personal user account (skipping process defined in Fig. 7). Similarly as illustrated in Fig. 6, the violator sends a message to the system. But this time the message includes a spy program (*data flow* Message containing a spy program), which is started after the message is accepted (*task* Accept message) and read (*task* Read message). The spy program initialises a new task (e.g., Extract data from database), which sends an inquiry to the database and extracts the Usernames and Passwords of existing users. These data are then attached to a reply message, which is sent to the violator (*task* Sends an answer and *data flow* Demand for registration + data copied from database).

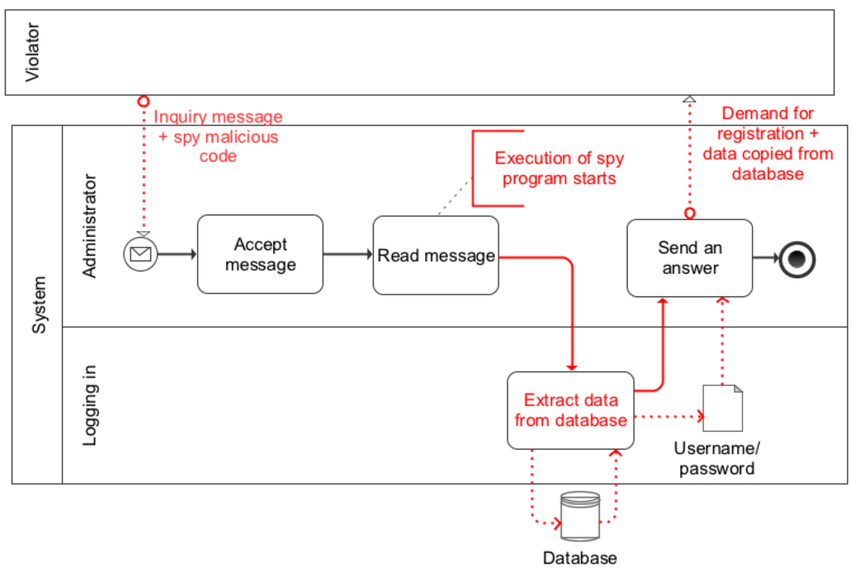


Fig. 8. Message Handling Process Including Security Risk Attack

In this analysis we are able to identify the ISSRM *threat agent* (e.g., Violator) and the ISSRM *attack method* (e.g., Message containing a spy program and Extraction of data from the database). Combination of these elements forms a security *threat*. The direct impact of this threat is that the *confidentiality* of the Usernames and Passwords is broken. On the other hand, this ISSRM *impact* provokes another *impact*, which negates the *integrity of the business processes*; i.e., the Violator is able now to access the system without registering, and, thus, change the business processes according to his needs.

Risk treatment involves deciding how the identified security flows could be mitigated. In our example we will take a *risk reduction* – i.e., actions to lessen the probability of the negative consequences – decision.

Security requirements definition. To reduce the probability of accepting the message, which contains a spy program, firstly, we introduce a *task* for Message scanning, as defined in Fig. 9. If scanning of the message reports a problem, the message is deleted and the message sender is blocked (*task* Block user/Delete message). Secondly, another security requirement includes the *task* Control activity of DB access. If there is a try to access the Database during the message handling process, it is blocked (*task* Block DB access). The final security requirement includes control of the outgoing/sent information (*task* Out-coming traffic control). This investigates if the response message is of the same length as initially defined. If this check reports a problem, the system stops the message sending (*cancel end event* Operation stopped).

Control implementation. The BPMN application is typically performed at the system analysis stages. Thus, implementation of the security requirements remains postponed for the later system development stages. On the other hand the iteration of the ISSRM process is needed where the current security requirements (e.g., ones introduced in Fig. 9) would be investigated for the new security risks.

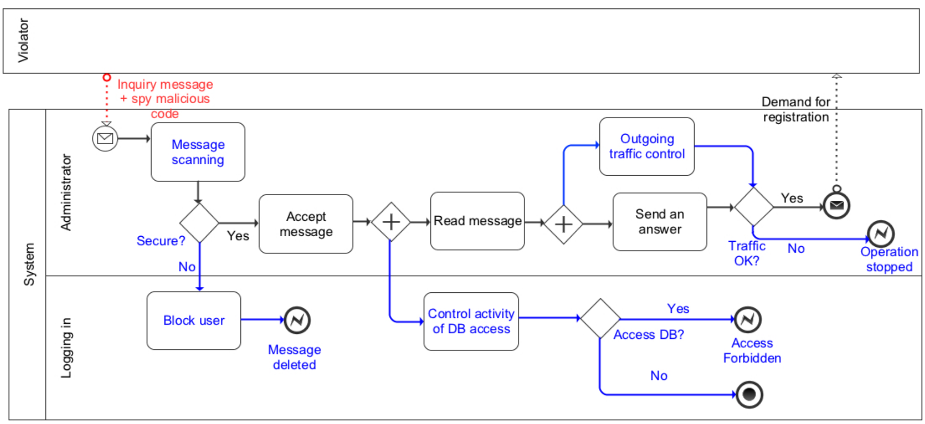


Fig. 9. Message Handling Process Including Security Requirements

4 ISSRM and BPMN Alignment

The running example illustrates a semantic alignment between ISSRM and BPMN. We show how BPMN is applied to consider possible attack scenarios and how countermeasures are defined. We summarise this discussion in Table 1.

Asset-Related Concepts. As described in Section 2, the ISSRM *business asset* could include valuable processes and information. In the first place the BPMN approach is meant for describing business processes within organisation. Thus, we can observe its constructs, such as *task, gateway*, *event* and their connecting link, i.e., *sequence flow*, that they help describing valuable processes. In the BPMN model the *flow objects* (i.e., *task, gateway* and *event*) are contained in the BPMN *containers*; i.e., *pools* and *lanes*. In other words the *container* constructs support definition and execution of the *business processes*. In terms of ISSRM, we align the *pool* and *lane* constructs to the ISSRM *information system assets*. The BPMN *data object*, which describes the required or produced data, is aligned to the ISSRM *business asset*, and BPMN *data store* is defined as ISSRM *IS asset*.

Table 1. Alignment of the ISSRM Concepts and the BPMN Constructs

The ISSRM domain model		BPMN constructs	Example
Asset-related concepts	Asset	–	–
	Business asset	*Data object*; *Task*, *Gateway*, *Event*, *Sequence flow*	Username and Password; Processes of Message handling, User registration, and User login to the system
	IS asset	*Data store* *Pool, Lane*	Database; System, Database connection, Message
	Security criterion	–	*Confidentiality* of Usernames and Password; *Integrity* of processes for Message handling, User registration and User login to the system
Risk-related concepts	Risk	–	–
	Impact	–	Confidentiality of Usernames and Password is broken; Integrity of processes is negated
	Event	–	–
	Threat	A combination of constructs for *Threat agent* and *Attack method*	A violator sends a message containing a spy program, which extract info from database and sends it back to the violator.
	Vulnerabilit y	–	Message is being handled without any scanning; The outgoing traffic is not monitored; The access to database is not controlled
	Threat agent	*Pool*	Violator
	Attack method	*Task*; *Flows (e.g., Data flow* with the label describing attack method; *Data association flow* with the label describing attack method);	Extract info from database; *Data flow* Message containing a spy program; *Data association flows* Sends a request and Gets data
Risk treatment related concepts	Risk treatment	–	Reduction (but other decision are also possible)
	Security requirement	*Task*, *Gateway*, *Event*, *Sequence flow*	*Tasks* Message scanning; Block user/Delete message; Control activity of DB access; Block DB access; Stop operation; Outgoing traffic control *Gateways* Secure?; Access to DB?; Traffic ok? *Events* Message deleted; Access forbidden; Operation stopped
	Control	–	–

The BPMN approach does not contain any constructs for explicit definition of the ISSRM *security criterion*. However, the created model can suggest the implicit expression (e.g., *Confidentiality of username and password*; *Integrity of the process*).

Risk-related concepts present what major principles should be taken into account when defining the potential risks. In principle the BMPN does not have the direct means to model security risks. However, in our example we have applied BPMN to model the negative and harmful processes. We have observed that the BPMN *pool*, when represents a negative/not intended actor, could be characterised as the ISSRM *threat agent*. Thus, the means that the *threat agent* is capable to use, are considered as the ISSRM *attack method*. For example, the BPMN *task*, as an atomic activity, when initialised by the "non-intended" actor, should be understood as the "means by which a threat agent executes threat"; such a *task* is aligned to the ISSRM *attack method*. Similar argumentation could be done about the BPMN *flow* and *data association flow*, which are also aligned to the ISSRM *attack method*.

We have not identified any explicit BPMN constructs to model the ISSRM *risk*, *impact*, *event*, or *vulnerability*. But we have observed that some of these concerns could be identified implicitly from the analysed problem. For instance, we can describe the ISSRM *threat* as the combination of the *threat agent* and *attack method* (see Table 1). Furthermore, two system *vulnerabilities* (namely, Message is being handled without any scanning and The outgoing traffic is not monitored) are identified. The third *vulnerability* (i.e., The access to database is not controlled) is found regarding the *database*. Finally, we can also define implicitly the ISSRM *impact*, which constitutes the negation of the identified *security criteria* and harm to the corresponding *assets*. These implicitly identified examples could not be expressed with the BPMN constructs.

Risk treatment-related concepts describe the decisions that should be taken, and controls to be implemented in order to mitigate the identified risks. In our example we select the *risk reduction*. However, other types of ISSRM *risk treatment decision* could also be taken depending on the level of risks mitigation.

The ISSRM *security requirements* are presented using the BPMN *task*, *gateway*, and *event* constructs connected using *sequence flow* links. For instance, the *security requirement* to mitigate the vulnerability Message is being handled without any scanning, starts with the BPMN *task* Message scanning, followed by the *gateway* Secure?. If the problem is found the *task* Block user/Delete message, and the process finishes with the *event* Message deleted. We do not align any BPMN construct to the ISSRM *controls*. However, we should note that in late system development stages the combination of the BPMN *task*, *gateway*, and *event* constructs (as illustrated above) might result in different security control modules.

5 Discussion and Conclusion

Our major contribution is the semantic alignment of the BPMN constructs to the ISSRM concepts. In addition we define a way to elicit security requirements for the important business processes. In this section we discuss validity, conclude the study with the potential BPMN extensions, and present the related and future work.

5.1 Threats to Validity

Our results contain a certain degree of subjectivity. Two researchers have performed this study. Thus, it might mean that some aspects of the BPMN approach or its application could be interpreted and aligned to the ISSRM concepts differently. Also, the running example involves the subjective decisions on how problem needs to be modelled. For instance, we have taken a risk reduction decision. The security requirements would be different if one would take the risk avoidance decision.

The scope of the current work is limited to the BPMN *descriptive modelling*. We acknowledge the importance to investigate the *analytical* and *executable* modelling, but this remains for the future research. Finally, in this work we analyse only a simple example of the Internet store. Although this example is realistic, we have not applied it in the practical settings. Thus, our analysis remains based on the selected BPMN literature [18] [21] [24].

5.2 BPMN Extensions towards Security Risk Management

In general, the BPMN approach is not specifically dedicated to the security modelling but to the business process modelling. On one hand we argue that the major version of the language should not loose its original purpose, and it should remain relatively simple. On the other hand we illustrate that BPMN provided the major set of constructs that help understanding important business assets, their security risks, and potential security requirements. Certainly this requires some potential language extensions:

- Using BPMN we are able to address only a part of the ISSRM domain model. For example, we were not able to express the ISSRM *security criterion*, *risk*, *impact*, *vulnerability*, *risk treatment*, and *control* constructs. This situation suggests potential extensions of the BPMN approach (at the concrete syntax, abstract syntax and semantic levels) and this is a potential direction for future research.
- The same constructs used for different ISSRM concepts. This could be noticed for the BPMN *task*, which is used to express the ISSRM *business asset*, *attack method*, and *security requirement* constructs; the BPMN *pool*, which helps modelling the ISSRM *threat agent* and *IS asset* constructs; and also some other constructs and links. This situation might provoke a readability and comprehensibility problem. There might be few solutions. The modellers could apply meta-labelling to identify different ISSRM-related concepts (e.g., [Business asset], [Attack method], or [Security requirement]) or introduce differentiating variables (e.g., *white* for the *asset-related*, *red* for the *risk-related*, and *blue* for the *treatment-related* constructs) between the same BPMN constructs aligned to different ISSRM constructs.

During our analysis we faced with a problem when one ISSRM concept could be presented using several BPMN constructs. For example, the ISSRM *security requirement* is modelled using the combination of the ISSRM *task*, *gateway*, *event* constructs and *sequence flow* links. This makes it difficult to understand the heuristics of the modelling process. Thus, it could be helpful to define rules and/or patterns to guide the use of the (security) modelling constructs.

5.3 Related Study on Security-Oriented BPMN

In [19] Rodríguez *et al.* propose the BPMN extensions for modelling secure business processes through understanding the security requirements. Firstly, their proposal illustrates the extension of the BPMN abstract syntax with the security-related concepts such as non-reputation, attack harm detection, integrity, privacy, access control, security role and security permission. Secondly, the concrete BPMN syntax is extended through the stereotypes introduced to the ordinary constructs of BPMN. The study does not include any consideration of the extension semantics. Further, in [20] some extensions of BPMN (called *BPSec*) are proposed towards the graphical representation of security requirements. They present a symbol of *padlock* to express security requirements and a *padlock with twisted corner* for audit register.

In [16] Menzel *et al.* proposes the BPMN enhancements towards trust modelling. They focus on the outline the metric that describes the value of enterprise assets and pay attention to the level of security or so called trust level of each participant of the process. Here, enterprise assets are presented using BPMN tasks, data objects, and communication links between tasks and participants. Authors define how to enable trustworthy interactions, organisational trust, and security intensions through BPMN. Other proposed extension is a security policy model used to define specific security patterns for authorisation, authentication, integrity, and confidentiality.

The limitations of these works [16] [19] [20] are that they focus either on a coarse-grained level, or target only some security aspects in business processes. In comparison our study does not propose any BPMN extensions. However, we present a semantically grounded fine-grained analysis based on the well-established ISSRM domain model [7] [15]. As a result we present the alignment between ISSRM concepts and the BPMN constructs, which allows developers to understand current BPMN means to deal with security. Also we identify potential BPMN extensions towards security both at the (concrete and abstract) syntax and at the security risk-oriented semantics levels. In other words we explore the reasons *why* and *how* BPMN needs to be extended to consider security at the business process modelling.

Paja *et al*. introduce a method to understand *security needs* through participants' objectives and interactions [17]. *Security requirements* are captured in terms of social commitments between the actors of the system. Then these security requirements are used to annotate business processes modelled in BPMN. Similarly, in our proposal we argue that security annotated BPMN models could be further analysed using the same modelling language, namely BPMN. The advantage is that the business analyst would not be required to learn yet another modelling notations, but would be able systematically reason for the return on security investment in business processes.

5.4 Related Studies on Security Risk-Oriented Modelling Languages

BPMN is not the only language assessed for the IS security risk management: ISSRM has been used to evaluated Secure Tropos [14], misuse cases [13], KAOS extensions to security [15], and Mal-activity diagrams [5]. But BPMN is the language to define the business process modelling. We have not found any business modelling language, which would support security analysis; thus the recent standard [24] for business process modelling was our natural choice. We envision that after analyzing a number of languages for security modelling it will be possible to facilitate model transformation and interoperability between them, thus introducing the security analysis from the early development stages to design and implementation, also resulting in a sustainable and secured system. Such a model transformation would be supported by transformation rules, developed on the semantic alignment of the (*business* and *security*) modelling approaches to the common base, i.e., the ISSRM domain model. However, definition of these transformation rules also remains a future study.

References

1. Alberts, C.J., Dorofee, A.J.: OCTAVE Method Implementation Guide Version 2.0. Carnegie Mellon University - Software Engineering Institute, Pennsylvania (2001)
2. Asnar, Y., Giorgini, P., Massacci, F., Zannone, N.: From Trust to Dependability through Risk Analysis. In: Proceedings of ARES 2007, pp. 19–26. IEEE Computer Society (2007)
3. AS/NZS 4360, Risk management. SAI Global (2004)
4. Braber, F., Hogganvik, I., Lund, M.S., Stølen, K., Vraalsen, F.: Model-based Security Analysis in Seven Steps—a Guided Tour to the CORAS Method. BT Technology Journal 25(1), 101–117 (2007)
5. Chowdhury, M.J.M., Matulevičius, R., Sindre, G., Karpati, P.: Aligning Mal-activity Diagrams and Security Risk Management for Security Requirements Definitions. In: Regnell, B., Damian, D. (eds.) REFSQ 2011. LNCS, vol. 7195, pp. 132–139. Springer, Heidelberg (2012)
6. Common Criteria version 2.3, Common Criteria for Information Technology Security Evaluation, CCMB-2005-08-002 (2005), `http://www.tse.org.tr/turkish/belgelendirme/ortakkriter/ccpart2v2.3.pdf`
7. Dubois, E., Heymans, P., Mayer, N., Matulevičius, R.: A Systematic Approach to Define the Domain of Information System Security Risk Management. In: Intentional Perspectives on Information Systems Engineering, pp. 289-306. Springer (2010)
8. Firesmith, D.G.: Engineering Safety and Security Related Requirements for Software Intensive Systems. In: Companion to the Proceedings of the 29th International Conference on Software Engineering (COMPANION 2007), p. 169. IEEE Computer Society (2007)
9. Haley, C.B., Laney, R.C., Moffett, J.D., Nuseibeh, B.: Security Requirements Engineering: A Framework for Representation and Analysis. IEEE Transactions on Software Engineering 34, 133–153 (2008)
10. Herrmann, A., Morali, A., Etalle, S., Wieringa, R.: Risk and Business Goal Based Security Requirement and Countermeasure Prioritization. In: Niedrite, L., Strazdina, R., Wangler, B. (eds.) BIR Workshops 2011. LNBIP, vol. 106, pp. 64–76. Springer, Heidelberg (2012)
11. ISO/IEC Guide 73, Risk management - Vocabulary - Guidelines for use in standards. International Organization for Standardization, Geneva (2002)
12. Jürjens, J.: Secure Systems Development with UML. Springer, Heidelberg (2005)
13. Matulevičius, R., Mayer, N., Heymans, P.: Alignment of Misuse Cases with Security Risk Management. In: Proceedings of ARES 2008, pp. 1397–1404. IEEE (2008)
14. Matulevičius, R., Mayer, N., Mouratidis, H., Martinez, F.H., Heymans, P., Genon, N.: Adapting Secure Tropos for Security Risk Management in the Early Phases of Information Systems Development. In: Bellahsène, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 541–555. Springer, Heidelberg (2008)
15. Mayer, N.: Model-based Management of Information System Security Risk. Doctoral Thesis, University of Namur (2009)

16. Menzel, M., Thomas, I., Meinel, C.: Security Requirements Specification in Service-oriented Business Process Management. In: ARES 2009, pp. 41–49 (2009)
17. Paja, E., Giorgini, P., Paul, S., Meland, P.H.: Security Requirements Engineering for Secure Business Processes. In: Niedrite, L., Strazdina, R., Wangler, B. (eds.) BIR Workshops 2011. LNBIP, vol. 106, pp. 77–89. Springer, Heidelberg (2012)
18. Remco, M., Dijkman, R.M., Dumas, M., Ouyang, C.: Formal Semantics and Analysis of BPMN Process Models using Petri Nets. Queensland University of Technology, TR (2007)
19. Rodríguez, A., Fernández-Medina, E., Piattini, M.: A BPMN Extension for the Modeling of Security Requirements in Business Processes. IEICE – Transactions on Information and Systems E90-D(4), 745–752 (2007)
20. Rodríguez, A., Fernández-Medina, E., Piattini, M.: UbiComp 2007. LNCS, vol. 4717, pp. 408–415 (2007)
21. Silver, B.: BPMN Method and Style: A Levels-based Methodology for BPMN Process Modeling and Improvement using BPMN 2.0. Cody-Cassidy Press (2009)
22. Stoneburner, G., Goguen, A., Feringa, A.: NIST Special Publication 800-30: Risk Management Guide for Information Technology Systems. National Institute of Standards and Technology, Gaithersburg (2002)
23. Trendowicz, A.: Tutorial: CoBRA - Cost Estimation, Benchmarking and Risk Analysis Method (2005), http://www.dasma.org/metrikon2005/tutorial_cobra.pdf
24. White, S.A.: Introduction to BPMN, IBM (2004), http://www.bpmn.org/Documents/Introduction_to_BPMN.pdf

A UML Extension for the Model-Driven Specification of Audit Rules

Bernhard Hoisl[1,2] and Mark Strembeck[1,2]

[1] Institute for Information Systems and New Media, Vienna University of Economics and Business (WU Vienna), Augasse 2-6, 1090 Vienna, Austria
[2] Secure Business Austria Research (SBA Research), Favoritenstrasse 16, 1040 Vienna, Austria
{bernhard.hoisl,mark.strembeck}@wu.ac.at

Abstract. In recent years, a number of laws and regulations (such as the Basel II accord or SOX) demand that organizations record certain activities or decisions to fulfill legally enforced reporting duties. Most of these regulations have a direct impact on the information systems that support an organization's business processes. Therefore, the definition of audit requirements at the modeling-level is an important prerequisite for the thorough implementation and enforcement of corresponding policies in a software system. In this paper, we present a UML extension for the specification of audit properties. The extension is generic and can be applied to a wide variety of UML elements. In a model-driven development (MDD) approach, our extension can be used to generate corresponding audit rules via model transformations.

Key words: Audit, Model-driven Development, UML.

1 Introduction

In information system security, an audit process records and analyzes data about the activities in a software system in order to detect security violations or to identify the cause of such violations (see, e.g., [1, 2, 3, 4, 5]). In this paper, we use the term *audit* for an "independent review and examination of records and activities to assess the adequacy of system controls and ensure compliance with established policies and operational procedures" [6]. Audit requirements not only stem from organization-specific management decisions or cost controlling policies, but also from corresponding laws and regulations, such as the Basel II Accord or the Sarbanes-Oxly Act (SOX) (see [7, 8]).

An audit process may involve different departments or divisions and focus on different assets of an organization, for example, financial records, customer privacy regulations, or access control policies. Nevertheless, all audit processes have in common that they are more and more based on and supported through information systems. For this reason, the software systems of an organization must be able to keep an audit trail of all audit-relevant business processes and activities. However, process modeling languages such as BPMN [9] or UML activity

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 16–30, 2012.

diagrams [10] do not provide native language elements to model such security properties. Thus, in order to properly enforce business-level security concerns in the corresponding software systems we need to integrate these concepts in a modeling language.

In recent years, model-driven development (MDD; see, e.g., [11, 12]) emerged as an approach for the specification of tailored domain-specific software systems. Due to its versatility, MDD can be applied as an approach for the systematic specification of information system security properties (see, e.g., [13, 14, 15, 16]). In the context of MDD, domain-specific languages (DSLs) are tailor-made (computer) languages for a specific problem domain (see, e.g., [17, 18, 19]). In general, a DSL can be defined as a standalone language or as a domain-specific extension to a pre-existing (modeling or programming) language. Such domain-specific extensions are also called "embedded DSLs".

In this paper, we present an approach for modeling system audits. In particular, we present a domain-specific UML extension that provides new language elements for the specification of audit events, audit rules, and notifications (or actions) that are triggered via audit events. The remainder of this paper is structured as follows: in Section 2 we give an overview of our audit modeling approach. Section 3 describes the metamodel, syntax, and semantics of our UML extension. Subsequently, Section 4 gives an example how our extension can be used to describe different audit modeling perspectives. After that, Section 5 summarizes related work and Section 6 concludes the paper. In addition, Appendix A provides a textual concrete syntax for our UML extension.

2 Motivation and Approach Synopsis

For each organization, a number of laws, regulations, and internal rules demand that the organization records certain activities or decisions which have a direct impact on the corresponding information systems (see, e.g., [20, 21, 22]). In particular, audit trails are needed to discharge an organization's reporting duties, for example, to prove the correctness of certain financial transactions (such as the enforcement of the four-eyes-principle for procurement operations). However, software engineers are usually not aware of all legal requirements that must be fulfilled by a software system. Therefore, we need a means to incorporate audit requirements in the respective software models. On the one hand, such a means should support the software engineer to model corresponding audit properties in a standard modeling language (such as the UML), on the other hand it should facilitate the communication between software engineers and domain experts (such as lawyers or experts from a certain business domain).

Moreover, because software systems as well as laws and regulations change over time, an extension for audit modeling should support the integration of audit properties with many different types of (heterogeneous) systems. Synchronous request/reply communication typically results in a strong coupling of interacting components. In contrast to that, a loose coupling of software services helps to integrate many different types of heterogeneous (legacy) systems (see, e.g., [23]).

Event-based communication is an important paradigm to model and implement such loosely-coupled systems—it is asynchronous and inherently decouples interacting system components (see, e.g., [24]). Event-based communication follows a publish/subscribe scheme where software components can produce and consume events. This means, an event producer does neither know the consumers of its events, nor does the producer publish events with the intention to trigger some action in an other component. Therefore, event-based components only have to know how to react on a particular notification and then publish events to "whom it may concern". This allows for a straightforward integration of new components and, thus, directly supports the evolution of event-based systems. Moreover, because event producers and event consumers are almost completely decoupled, event-based components are widely independent of each other which, again, makes these components more easy to adapt and extend.

In this paper, we, therefore, present an approach for the event-based modeling of audit properties. Fig. 1 shows an informal overview for the main conceptual elements of our approach. In essence, we provide a UML extension to model audit properties of software artifacts that can be applied to different types of UML models. We have chosen the UML because it is the de-facto standard for modeling information systems and provides native support for all types of software models as well as for event-based modeling. The audit properties defined via this modeling extension can then be used to generate corresponding audit rules that can be enforced in a software system.

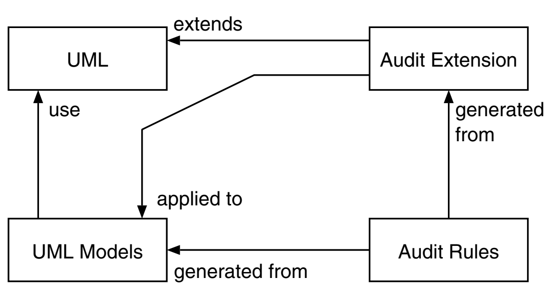


Fig. 1. Audit extension for UML models

Our extension supports the definition of different perspectives, each of which models a particular aspect of system audits (see Fig. 2). Subsequently, model transformations (see, e.g., [25, 26, 27]) can be used to generate different types of software artifacts and audit rules from these models. The generated artifacts then enforce the behavior that was defined on the modeling level. Thereby, our UML extension allows to map audit requirements from the modeling- to the system-level. Because the UML provides an integrated family of modeling notations, a UML extension helps to avoid the semantic gap that could occur if we integrate models that are defined in different modeling languages (see, e.g., [28, 29]).

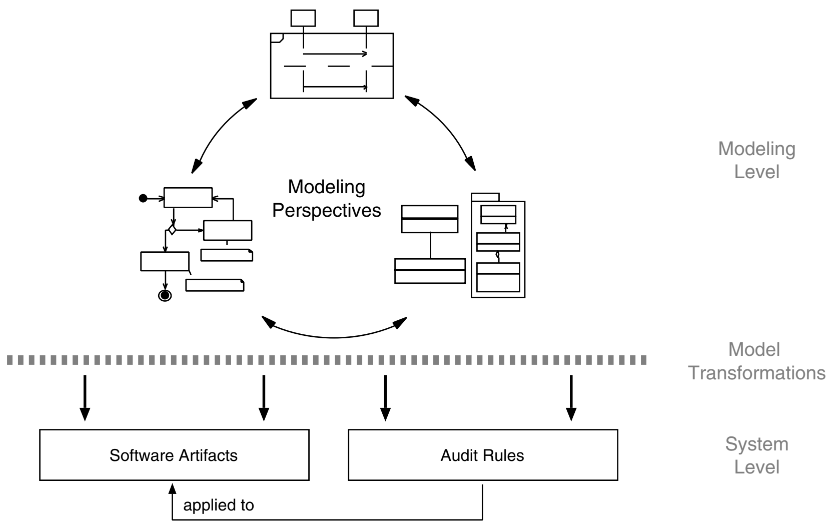


Fig. 2. Modeling-level audit properties are transformed into system artifacts

Our extension is generic and allows to define audit requirements for arbitrary elements in arbitrary UML models. Moreover, it is event-based and thereby enables a loose-coupling and a straightforward integration with different types of (heterogeneous) software components.

3 UML Audit Extension

3.1 Metamodel Overview

In this section, we specify a UML extension (see Fig. 3) for modeling event-based audit requirements. In particular, we introduce a new package called *SecurityAudit* as a UML metamodel extension [10]. The package consists of both, a UML stereotype specialization and MOF-based (Meta Object Facility, [30]) extensions.

In general, the UML can be extended in two ways: (1) by using UML profiles [10] or (2) by introducing new modeling concepts on the metamodel level. UML profiles provide a mechanism for the extension of existing UML metaclasses to adapt them for non-standard purposes. However, UML profiles are not a first-class extension mechanism (see [10, page 660]). They extend existing metaclasses of the UML metamodel and the extension defined through a profile must be consistent with the semantics of the extended (original) UML metaclasses. For this reason, more complex extensions are defined on the level of the UML metamodel (see [10, 30]). An extension of the UML metamodel allows to define new and specifically tailored UML elements (defined via new metaclasses), and allows to define a customized notation, syntax, and semantics for the new modeling elements. In our extension, we employ a combination of both methods to take advantage of each mechanism.

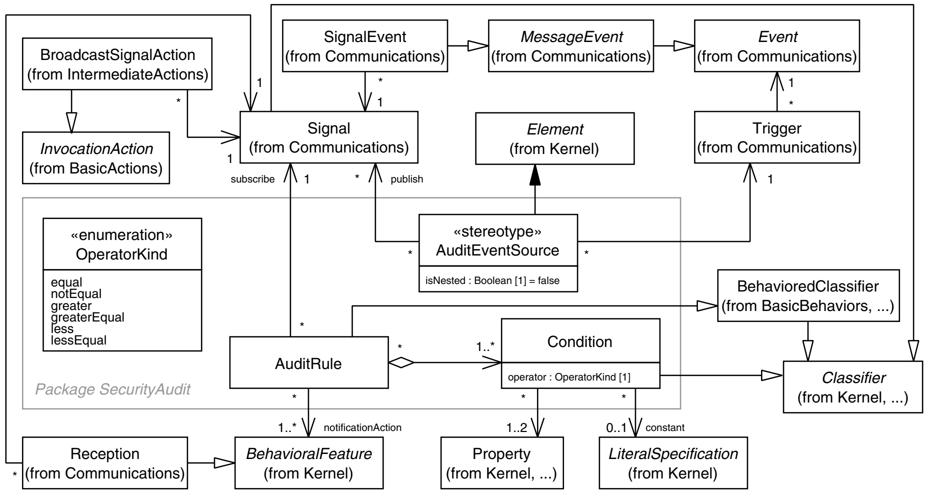


Fig. 3. UML extension for modeling event-based audit requirements

In our extension, the `«stereotype» AuditEventSource` extends the UML `Element` metaclass (see Fig. 3). As a specialized `Element` stereotype, it is possible to define any UML element as being the source for an event that may trigger an audit-related behavior execution. In this way, an integration with arbitrary (pre-existing) UML models is possible. The `isNested` attribute defines whether the `AuditEventSource` stereotype is applied to the owned elements of a stereotyped element (e.g. to all nodes in an UML activity). Hence, it is possible to tag the owner element only and recursively apply the `AuditEventSource` stereotype and its properties to all nested elements.

A `Trigger` relates an `Event` to a certain type of `Signal` that is published each time this particular event occurs. A UML `Signal` is a specialized `Classifier` and can carry data which is passed via the corresponding send invocation occurrence. Events are published through a corresponding `BroadcastSignalAction` which transmits a `Signal` instance to all potential target objects in a system (see also Fig. 3 and [10]). We use a `BroadcastSignalAction` in favor of a `SendSignalAction` because events are published independent of the entities (software components) consuming the events (see, e.g., [24]).

Modeling the receipt of a `Signal` instance is done via an `AcceptEventAction` (in behavior diagrams) or via the `Reception` element (in structure diagrams). Either way, a `SignalEvent` represents the receiving of an asynchronous `Signal` instance. The elements modeling the transmission and receipt of `Signal` instances act as the underlying event notification service, which mediates between notification producers and consumers (according to the publish/subscribe pattern; see, e.g., [24, 31]).

An `AuditRule` is defined as a specialized `BehavioredClassifier` and is subscribed to a specific `Signal` (see Fig. 3). Each `AuditRule` consists of one or more `Condition` elements. A `Condition` evaluates a certain attribute of a `Signal` and checks the corresponding attribute value (e.g. by using binary infix operators, as in: "price $<$ 63.50" or "currency $=$ EUR"). In our extension, a `Condition` can test either two `Properties` against each other, or it can check a `Property` against a pre-defined constant value (a `LiteralSpecification`). A UML `LiteralSpecification` references an instance of a primitive data type[1]. For basic condition matching, the `«enumeration» OperatorKind` specifies an exemplary list of valid self-explanatory operator alternatives. Note, however, that these infix comparison operators can easily be extended to represent other types of operators, for instance, n-ary prefix operators (such as `isInAscendingOrder(...)`, `isInDescendingOrder(...)`, or `includes(...)`).

An `AuditRule` matches an event (resp. the corresponding `Signal`) if all `Conditions` that are associated with this `AuditRule` are fulfilled. In case all `Conditions` of an `AuditRule` are fulfilled, the respective `AuditRule` triggers the execution of a certain `BehavioralFeature` (see Fig. 3). This `BehavioralFeature` implements a notification action that informs another system entity that one of the audit rules was activated and causes a certain behavior (e.g., generating a new log entry in the audit trail).

In general, every stereotype must be included (directly or indirectly) in a profile [10]. For our extension, we define that the `«stereotype» AuditEventSource` is contained in the `AuditEventSourceProfile`. We use the Object Constraint Language (OCL, [32]) to formally specify constraints for our modeling extension:

```
context AuditEventSource inv:
  self.profile.name = 'AuditEventSourceProfile'
```

As this profile is an integral part of our extension, we define that it must be applied to the package `SecurityAudit`:

```
context SecurityAudit inv:
  self.profileApplication->exists(
    appliedProfile.name = 'AuditEventSourceProfile')
```

The relationship of the `SecurityAudit` package, its profile application, and their referenced metamodels are shown in Fig. 4. The profile `AuditEventSourceProfile` references the UML metamodel and is applied to the package `SecurityAudit`. As we define the package `SecurityAudit` via a UML metamodel extension, it references the MOF and uses elements from the UML. The MOF is self-describing (through reflection; see [30]) and, therefore, does not need another metamodel for its specification. Furthermore, the MOF specification reuses modeling constructs from the UML infrastructure library (through package imports; see [33]).

---

[1] The UML defines six `LiteralSpecification` subtypes: `LiteralNull`, `LiteralBoolean`, `LiteralInteger`, `LiteralReal`, `LiteralString`, and `LiteralUnlimitedNatural` [10]. Due to space limitations these six specializing `LiteralSpecifications` are omitted in Fig. 3.

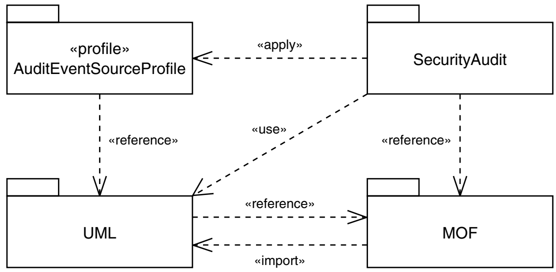
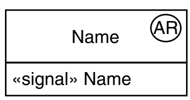
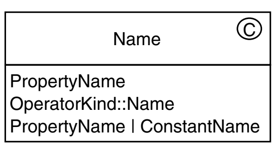


Fig. 4. Dependencies of the `SecurityAudit` package

Table 1. Modeling elements of the `SecurityAudit` package

Node type	Notation	Explanation
AuditRule		An `AuditRule` is shown as a rectangle with the encircled characters *AR* in the upper right corner. The optional `Signal` compartment states that the `AuditRule` is prepared to react to the receipt of a certain signal (see [10]).
Condition		A `Condition` is shown as a rectangle with the encircled character *C* in the upper right corner. The lower compartment includes the attributes and the operator that constitute the respective condition. The first attribute is the name of a `Property` which references a certain `Signal` attribute, the second attribute may either be another `Property` or a constant value (i.e. a `LiteralSpecification`), and the operator is of type `OperatorKind` (see Fig. 3). Thereby, a condition consists of an operator that compares two operands (for example "price $<$ 63.50" or "currency $=$ EUR").

3.2 Metamodel Elements' Syntax and Semantics

Table 1 shows the notation elements of the `SecurityAudit` package (see also Section 4). The other UML elements used in our examples correspond to the UML specification (see [10]).

In addition to the graphical modeling elements, Appendix A provides a textual syntax for event-based audits that is specified via a variant of the Backus-Naur-Form (BNF; see [34]). We have chosen the BNF as a context-free grammar as it is also applied in OMG specifications (e.g., [10, 32]), it is commonly used to formally specify the syntax of computer languages, and it is widely tool-supported (e.g., the Eclipse Xtext notation is very similar to an extended BNF). To model event-based audits, the graphical or the textual syntax can be used separately and equivalently. Moreover, it is also possible to combine the textual and graphical syntaxes (see the example in Section 4).

In addition, to the syntax definitions we specify OCL invariants that ensure the correct semantics of models defined with our UML extension (see Fig. 3). The `AuditEventSource` stereotype can be applied recursively to all owned elements of a tagged element (if the `isNested` attribute is set to `true`). All stereotype properties of the tagged owner element are inherited, except if a nested element explicitly defines its own `Trigger` and `Signal`. In this case, the properties of the tagged owner element are overwritten:

```
context AuditEventSource inv:
  self.isNested implies
    self.base_Class.ownedElement->forAll(oe |
      oe.getAppliedStereotype('AuditEventSourceProfile::
          AuditEventSource') <> null)
```

To be able to evaluate a `Condition` of an `AuditRule`, exactly one `Property` must be a referenced attribute of the subscribed `Signal` instance:

```
context AuditRule inv:
  self.condition->forAll(c |
    self.subscribe.ownedAttribute->intersection(
      c.property)->size() =
        c.ownedAttribute->select(oa |
          oa.name = 'property')->first().lowerBound())
```

We define that a `Condition` can test either two `Properties` against each other or one `Property` against a constant (as specified in the metamodel), but not both. Specifying a `Condition` without matching operands is also not allowed:

```
context Condition inv:
  self.property->size() + self.constant->size() =
    self.ownedAttribute->select(oa |
      oa.name = 'property')->first().upperBound().oclAsType(
          Integer)
```

Matching `Properties` against each other or against a `LiteralSpecification` constant implies that they conform to the same type (e.g., both are of type `<Primitive Type> Integer`):

```
context Condition inv:
  if self.constant->notEmpty() then
    self.property.type.conformsTo(self.constant.type)
  else
    self.property->forAll(p1,p2 |
      p1.type.conformsTo(p2.type))
  endif
```

4 Audit Modeling Perspectives

In this section, we describe an example for audit modeling of a simple event-based system. In order to thoroughly describe a software system, different modeling perspectives have to be defined. Therefore, we take different viewpoints into account to explain the application of our UML extension to different structural and behavioral models. The perspectives in Fig. 5 are exemplary and can be used interchangeable.

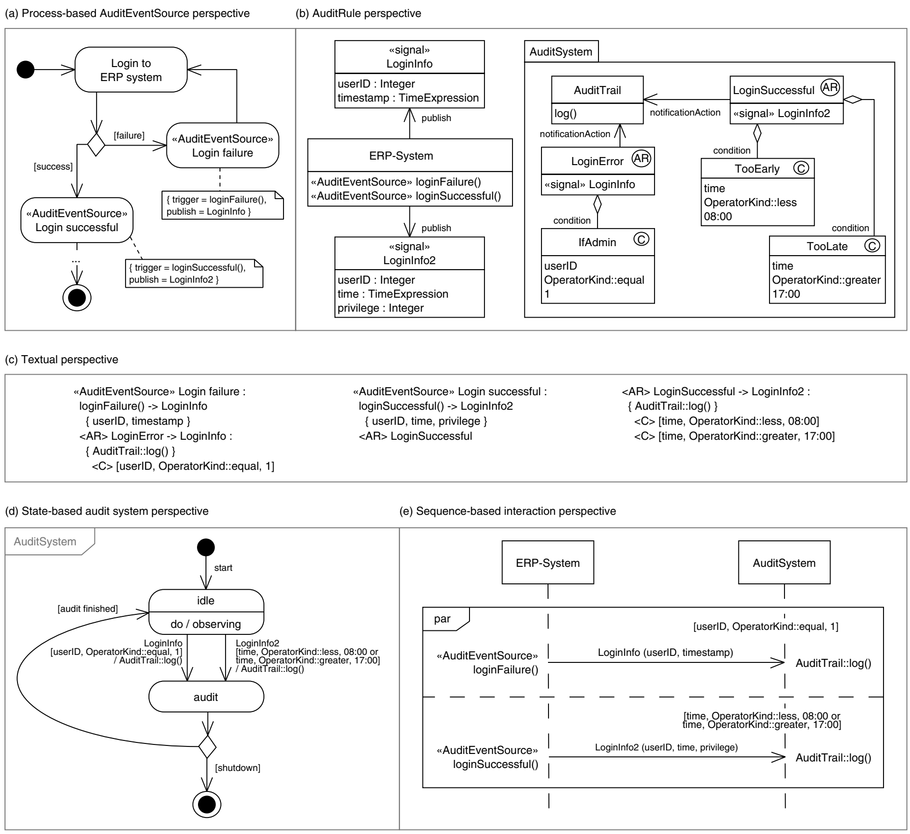


Fig. 5. Modeling event-based audit requirements from different perspectives

Fig. 5a shows a process-based perspective modeled via a UML activity diagram. Here, the `«AuditEventSource»` stereotype is applied to two `BroadcastSignal-Actions`. The example models a basic login process to an ERP system that should include audit trails for successful as well as for failed login attempts (indicated via the `«AuditEventSource»` stereotype). Two constraints are attached to the actions defining the `Trigger` for the audit event and the corresponding `Signal` classifier. However, using this perspective alone, information about the `Signals`, the `AuditRules`, their `Conditions` and `Actions` can not be modeled sufficiently.

Therefore, Fig. 5b presents the AuditRule perspective. It shows an ERP-System classifier that implements two methods which match the execution operations of the corresponding BroadcastSignalActions shown in Fig. 5a. The «AuditEvent-Source» stereotypes bind both, the «signal» LoginInfo to the loginFailure() method and the «signal» LoginInfo2 to the loginSuccessful() method. Furthermore, Fig. 5b shows two simple AuditRules LoginError and LoginSuccessful with each having a compartment defining the corresponding subscribed Signal. The AuditRule LoginError consists of one Condition (IfAdmin) which checks for failed administrator logins (i.e., if the userID included in the corresponding Signal instance equals 1[2]). The second AuditRule LoginSuccessful consists of two conditions which check if a login happened outside of normal business hours. If one of these Conditions evaluate to true, the log() method of the AuditTrail classifier is invoked (as both AuditRules reference the same notification action). This perspective, of course, omits all process information.

Fig. 5c shows an example of the textual perspective. The syntax conforms to the BNF grammar defined in Appendix A. The textual syntax is equivalent to the graphical AuditRule perspective (see Fig. 5b); i.e. all AuditRules and Conditions are equally defined. The textual syntax can be used complementary to the graphical representation.

Fig. 5d shows a perspective of the audit system as a UML state machine. The state machine is used to model the receiving Signal instances, their Conditions, and corresponding actions. As can be seen from the AuditRule and the textual perspective, the second Signal named LoginInfo2 serves as the notification message of action Login successful in the process-based view. The state machine, for instance, shows the same Signal, Condition, and action information associated with the corresponding transition. In this perspective, the modeled states and their transitions of an audit system reveal neither process- nor object-specific information.

Finally, Fig. 5e shows a message interaction perspective as a UML sequence diagram. Therein, the sending and receiving events of the two involved systems, together with the interchanged signal messages are shown. Both «AuditEventSource» events are defined for parallel execution, i.e. there is no sequential order between these events. The corresponding messages are defined via the respective Signal names including their owned attributes. The Conditions for invoking audit actions are defined as guards on the lifeline of the AuditSystem. This perspective neither shows the process flow nor the detailed structure of the audit rules.

All perspectives presented here are complementary and can be used interchangeable. The combination of perspectives are dependent on the modeled software system (e.g., state-based).

5 Related Work

In [35], Jürjens describes how to model audit security for smart-card payment schemes with UMLSec. The UMLSec extension is defined as a UML profile. Our

[2] For the sake of simplicity, we assume that the administrator of the ERP system has the value 1 for the attribute userID.

extension for audit modeling supports the definition of different audit perspectives and complements the UMLSec approach. In general, we extend the UML `Element` metaclass and, thereby, allow to extend a wide variety of UML elements with audit properties. Furthermore, our extension supports event-based modeling and, thus, aims to facilitate the integration of audit properties into pre-existing models for heterogeneous (or legacy) systems.

Rodríguez et al. [36] present a UML profile extension for activity diagrams which aims to support the specification of certain security properties (e.g., access control, integrity, non-repudiation, and privacy). In [36], audits are specified as an additional characteristic for another security property. The audit process is treated as a logging of data, and the logged data must be defined via attributes of the corresponding audited entity. In contrast, our extension is more generic and can be used to model audit rules for arbitrary UML elements. Moreover, our audit extension is integrated with other UML extensions for security modeling (see, e.g., [15, 37, 38, 39, 40])

In [41], Fernández-Medina et al. provide support for modeling access control and audit properties for multidimensional data warehouses with a UML profile definition. Audit requirements are considered by defining audit rules for logging user requests and activities. Audit rules are defined via a custom-made grammar specified in Extended Backus-Naur-Form (EBNF). These audit rules are represented in the form of constraints for a UML class diagram. In contrast, our approach is not specific to a particular application domain and can be integrated with other UML-based approaches.

In [42], an approach for the modeling of security-critical, service-oriented systems is presented. The authors provide a UML profile that defines stereotypes for the extension of class diagrams. Security patterns and protocols are applied to identified security critical use cases. Service composition rules can be defined as post-obligations to be taken into account while (or after) executing a protocol (e.g., auditing). In [42], audit requirements are not defined as specialized modeling elements, but via OCL constraints. Thus, the modeling approach is rather specialized and has a limited expressiveness (for both, syntax and semantics).

6 Conclusion

In this paper we presented a UML extension for modeling system audits. Our extension supports an event-based modeling style and thereby aims to enable the integration of audit properties in a wide variety of different types of UML models. We support the definition of structural and behavioral perspectives to model different aspects of system audits. In addition to graphical model elements, we also provide a fully equivalent textual syntax.

With our extension, each UML element can be defined as an audit event source. Thus, the extension is not limited to a specific type of UML diagram. Moreover, it can be customized to different types of system audits. However, in this paper we do not elaborate on the modeling of an event notification service (i.e., we omit `BroadcastSignalActions` and `AcceptEventActions` in our examples).

Furthermore, we neither show an example of nested audit models nor discuss wildcard triggers which invoke a specified audit rule on every event occurrence of an element or nested elements. Application-specific OCL constraints can be used to further refine, for instance, event triggers or audit rules (e.g., pre- and postconditions). The textual syntax of our extension is fully integrated with the graphical perspectives and can be applied either interchangeable or in addition to the graphical models.

In our future work, we will integrate support for the explicit modeling of composite as well as hierarchical audit event types. Moreover, we are working on a tool integration of our extension which will implement both, the graphical and textual syntax.

Acknowledgments. This work has partly been funded by the Austrian Research Promotion Agency (FFG) of the Austrian Federal Ministry for Transport, Innovation and Technology (BMVIT) through the Competence Centers for Excellent Technologies (COMET K1) initiative and the FIT-IT program.

References

1. Garera, S., Rubin, A.: An Independent Audit Framework for Software Dependent Voting Systems. In: Proc. of the 14th ACM Conference on Computer and Communications Security (CCS), pp. 256–265 (2007)
2. Hasan, R., Winslett, M.: Efficient Audit-based Compliance for Relational Data Retention. In: Proc. of the 6th ACM Symposium on Information, Computer and Communications Security, pp. 238–248 (2011)
3. King, J., Smith, B., Williams, L.: Modifying Without a Trace: General Audit Guidelines are Inadequate for Open-source Electronic Health Record Audit Mechanisms. In: Proc. of the 2nd ACM SIGHIT International Health Informatics Symposium, pp. 305–314 (2012)
4. Sandhu, R., Samarati, P.: Authentication, Access Control, and Audit. ACM Computing Surveys 28(1), 241–243 (1996)
5. Schneier, B., Kelsey, J.: Secure Audit Logs to Support Computer Forensics. ACM Transaction on Information and System Security 2(2), 159–176 (1999)
6. Committee on National Security Systems: National Information Assurance (IA) – Glossary (2010), http://www.cnss.gov/Assets/pdf/cnssi_4009.pdf
7. Basel Committee on Banking Supervision: Basel II: International Convergence of Capital Measurement and Capital Standards (2004), http://www.bis.org/publ/bcbs107.pdf
8. United States Congress: Sarbanes-Oxley Act of 2002 (2002), http://www.sec.gov/about/laws/soa2002.pdf
9. Object Management Group: Business Process Model and Notation (BPMN) – Version 2.0 (2011), http://www.omg.org/spec/BPMN/2.0/PDF
10. Object Management Group: OMG Unified Modeling Language (OMG UML), Superstructure – Version 2.4.1 (2011), http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF
11. Selic, B.: The Pragmatics of Model-driven Development. IEEE Software 20(5), 19–25 (2003)

12. Stahl, T., Völter, M.: Model-Driven Software Development. John Wiley & Sons (2006)
13. Basin, D., Doser, J., Lodderstedt, T.: Model Driven Security: From UML Models to Access Control Infrastructures. ACM Transactions on Software Engineering and Methodology (TOSEM) 15(1) (January 2006)
14. Hoisl, B., Sobernig, S.: Integrity and Confidentiality Annotations for Service Interfaces in SoaML Models. In: Proceedings of the International Workshop on Security Aspects of Process-aware Information Systems (SAPAIS). IEEE, Vienna (2011)
15. Strembeck, M., Mendling, J.: Modeling Process-related RBAC Models with Extended UML Activity Models. Information and Software Technology (IST) 53(5), 456–483 (2010)
16. Wolter, C., Menzel, M., Schaad, A., Miseldine, P., Meinel, C.: Model-driven business process security requirement specification. Journal of Systems Architecture 55(4) (April 2009)
17. Deursen, A.V., Klint, P.: Little Languages: little Maintenance? Journal of Software Maintenance: Research and Practice 10(2), 75–92 (1998)
18. Mernik, M., Heering, J., Sloane, A.: When and How to Develop Domain-specific Languages. ACM Computing Surveys (CSUR) 37(4), 316–344 (2005)
19. Strembeck, M., Zdun, U.: An Approach for the Systematic Development of Domain-Specific Languages. Software: Practice and Experience (SP&E) 39(15) (October 2009)
20. Cannon, J.C., Byers, M.: Compliance Deconstructed. ACM Queue 4(7) (September 2006)
21. Damianides, M.: How does SOX change IT? Journal of Corporate Accounting & Finance 15(6) (2004)
22. Mishra, S., Weistroffer, H.R.: A Framework for Integrating Sarbanes-Oxley Compliance into the Systems Development Process. Communications of the Association for Information Systems (CAIS) 20(1) (2007)
23. Hohpe, G., Woolf, B.: Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley, Boston (2004)
24. Mühl, G., Fiege, L., Pietzuch, P.: Distributed Event-Based Systems. Springer, Heidelberg (2006)
25. Mens, T., Gorp, P.V.: A Taxonomy of Model Transformation. Electronic Notes in Theoretical Computer Science 152, 125–142 (2006)
26. Sendall, S., Kozaczynski, W.: Model Transformation: The Heart and Soul of Model-Driven Software Development. IEEE Software 20(5) (2003)
27. Zdun, U., Strembeck, M.: Modeling Composition in Dynamic Programming Environments with Model Transformations. In: Löwe, W., Südholt, M. (eds.) SC 2006. LNCS, vol. 4089, pp. 178–193. Springer, Heidelberg (2006)
28. Axenath, B., Kindler, E., Rubin, V.: AMFIBIA: A Meta-Model for the Integration of Business Process Modelling Aspects. In: Leymann, F., Reisig, W., Thatte, S.R., van der Aalst, W. (eds.) The Role of Business Processes in Service Oriented Architectures. Number 06291 in Dagstuhl Seminar Proceedings (2006)
29. Zdun, U.: Patterns of Component and Language Integration. In: Manolescu, D., Voelter, M., Noble, J. (eds.) Pattern Languages of Program Design 5 (2006)
30. Object Management Group: OMG Meta Object Facility (MOF) Core Specification – Version 2.4.1 (2011), http://www.omg.org/spec/MOF/2.4.1/PDF/
31. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading (1995)
32. Object Management Group: OMG Object Constraint Language (OCL) – Version 2.3.1 (2012), http://www.omg.org/spec/OCL/2.3.1/PDF

33. Object Management Group: OMG Unified Modeling Language (OMG UML), Infrastructure – Version 2.4.1 (2011), http://www.omg.org/spec/UML/2.4.1/Infrastructure/PDF/
34. International Organization for Standardization: Information Technology – Syntactic Metalanguage – Extended BNF (ISO/IEC 14977) (1996), http://standards.iso.org/ittf/PubliclyAvailableStandards/s026153_ISO_IEC_14977_1996(E).zip
35. Jürjens, J.: Modelling Audit Security for Smart-Card Payment Schemes with UMLSEC. In: Proceedings of the 16th International Conference on Information Security, Paris, France (2001)
36. Rodríguez, A., Fernández-Medina, E., Trujillo, J., Piattini, M.: Secure Business Process Model Specification through a UML 2.0 Activity Diagram Profile. Decision Support Systems 51(3), 446–465 (2011)
37. Hoisl, B., Strembeck, M.: Modeling Support for Confidentiality and Integrity of Object Flows in Activity Models. In: Abramowicz, W. (ed.) BIS 2011. LNBIP, vol. 87, pp. 278–289. Springer, Heidelberg (2011)
38. Schefer, S., Strembeck, M.: Modeling Process-Related Duties with Extended UML Activity and Interaction Diagrams. In: Proc. of the International Workshop on Flexible Workflows in Distributed Systems, Electronic Communications of the EASST (March 2011)
39. Schefer, S., Strembeck, M.: Modeling Support for Delegating Roles, Tasks, and Duties in a Process-Related RBAC Context. In: Salinesi, C., Pastor, O. (eds.) CAiSE Workshops 2011. LNBIP, vol. 83, pp. 660–667. Springer, Heidelberg (2011)
40. Schefer-Wenzl, S., Strembeck, M.: Modeling Context-Aware RBAC Models for Business Processes in Ubiquitous Computing Environments. In: Proc. of the 3rd International Conference on Mobile, Ubiquitous and Intelligent Computing, MUSIC (June 2012)
41. Fernández-Medina, E., Trujillo, J., Villarroel, R., Piattini, M.: Access Control and Audit Model for the Multidimensional Modeling of Data Warehouses. Decision Support Systems 42(3), 1270–1289 (2006)
42. Memon, M., Hafner, M., Breu, R.: SECTISSIMO: A Platform-independent Framework for Security Services. In: Proceedings of the Modeling Security Workshop in Association with MODELS 2008, Toulouse, France (2008)

A Textual Syntax for the `SecurityAudit` Package

```
<SecurityAudit>      ::= (<AuditEventSource> | <AuditRule>)*
<AuditEventSource> ::= '<<AuditEventSource>>' UML::Element.name ':' <Trigger> '->' <Publication> <AuditRule>
<Trigger>            ::= UML::Trigger.name
<Publication>        ::= UML::Signal.name | <Signal>
<Signal>             ::= UML::Signal.name '{' UML::Signal.attribute.name [',' UML::Signal.attribute.name]* '}'
<AuditRule>          ::= '<AR>' UML::AuditRule.name ( '->' <Subscription> ': {' <Action> [',' <Action>]* '}' <Condition>+ )?
<Subscription>       ::= <Publication>
<Action>             ::= UML::BehavioralFeature.name
<Condition>          ::= '<C>' ( UML::Condition.name | '[' <Operand> ',' <Operator> ',' <Operand> ']' )
<Operand>            ::= <Property> | <Constant>
<Property>           ::= UML::Property.name
<Constant>           ::= UML::LiteralSpecification
<Operator>           ::= UML::OperatorKind::EnumerationLiteral
```

Applying Soft Computing Technologies for Implementing Privacy-Aware Systems

Christos Kalloniatis[1], Petros Belsis[2], Evangelia Kavakli[1], and Stefanos Gritzalis[3]

[1] Department of Cultural Technology and Communication, University of the Aegean, Harilaou Trikoupi & Faonos Str., 81100 Mytilene, Greece
chkallon@aegean.gr, kavakli@ct.aegean.gr

[2] Technological Education Institute of Athens, Department of Marketing, Agiou Spyridonos Street, 12210 Aigaleo, Athens, Greece
pbelsis@aegean.gr

[3] Information and Communication Systems Security Laboratory
Department of Information and Communications Systems Engineering
University of the Aegean, 83200, Samos, Greece
sgritz@aegean.gr

Abstract. Designing privacy-aware systems gains much attention in recent years. One of the main issues for the protection of users' privacy is the proper selection and realization of the respective Privacy Enhancing Technologies for the realization of the privacy requirements identified in the design phase. The selection of PETs must be conducted in a way that best fits the organization's needs as well as other organization's criteria like cost, complexity etc. In this paper the PriS method, which is used for incorporating security and privacy requirements early in the system development process, is extended by combining knowledge from a soft computing approach in order to improve the way that respective PETs are selected for the realization of the respective requirements incorporated during the design phase.

1 Introduction

A major challenge in the field of software engineering is to make users trust the software that they use in their everyday activities for professional or recreational reasons. Trusting software depends on various elements, one of which is the protection of user privacy. Protecting privacy is about complying with user's desires when it comes to handling personal information. Users' privacy can also be defined as the right to determine when, how and to what extend information about them is communicated to others.

Nowadays, protecting privacy is focused on reducing the information collected and stored to a minimum, and deleting the information as soon as it has served its purpose. Most of today's e-services are relying on stored data, identifying the customer, his preferences and previous record of transactions. However, combining such data will in many cases constitute an invasion of privacy.

Research efforts aiming to the protection of user privacy fall in two main categories: security-oriented requirement engineering methodologies and privacy

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 31–45, 2012.

enhancing technologies. The former focus on methods and techniques for considering security issues (including privacy) during the early stages of system development and the latter describe technological solutions for assuring user privacy during system implementation. The main limitation of security requirement engineering methodologies is that they do not link the identified requirements with implementation solutions. Understanding the relationship between user needs and the capabilities of the supporting software systems is of critical importance. Privacy enhancing technologies, on the other hand, focus on the software implementation alone, irrespective of the organizational context in which the system will be incorporated. This lack of knowledge makes it difficult to determine which software solution best fits the organizational needs. A review on a number of well-known security and privacy requirements engineering methods can be found in [1]. Due to limited space the comparison results are excluded from this paper but can be found in our previous work conducted in [1].

To this end, PriS, a new security requirements engineering method, has been introduced aiming to incorporate privacy requirements early in the system development process. PriS models privacy requirements in terms of business goals and uses the concept of privacy process patterns for describing the impact of privacy goals onto the business processes and the associated software systems supporting these processes.

The conceptual model of PriS uses a goal hierarchy structure. Every privacy requirement is either applied or not on every goal. The representation of a privacy requirement that constraints a goal is achieved by the use of a variable which can take two values, zero and one. If one of the privacy requirements is applied on a specific goal the respective privacy variable will be assigned with the value of one otherwise will remain zero which was also its initial value. Thus, on every privacy-related goal seven privacy variables are applied and representing which privacy requirements constraint the goal and which not (Since pseudonymity can be considered as part of anonymity, they are both addressed in one pattern). Following this way of working PriS ends up suggesting a number of implementation techniques based on the privacy requirements constraining the respective goals. While PriS successfully guides the developers through the implementation phase by suggesting a number of implementation techniques it fails to address the degree of participation of every privacy requirement for achieving the generic goal of privacy.

This paper applies the extended PriS (along with the soft computing approach) on an e-voting case study. Specifically, in section 2 the case study is presented. Section 3 presents a brief description of PriS along with its way of working. Section 4 presents the application of fuzzy PriS on the specific case study. Finally, section 5 concludes with pointers to future work.

2 PriS Conceptual Framework and Way of Working

As mentioned earlier, privacy enhancing technologies focus on the software implementation alone. In other words, there is no obvious link between the organizational processes that are constrained by the privacy requirements and the supporting software systems. This lack of knowledge makes it difficult not only to determine which software solution best fits the organizational needs but also to evaluate alternatives.

To this end, PriS provides a set of concepts for modeling privacy requirements in the organization domain and a systematic way-of-working for translating these requirements into system models. The conceptual model used in PriS, shown in figure 1, is based on the Enterprise Knowledge Development (EKD) framework [2,3], which is a systematic approach to developing and documenting organizational knowledge. This is achieved through the modeling of: (a) organizational goals, that express the intentional objectives that control and govern its operation, (b) the 'physical' processes, that collaboratively operationalise organizational goals and (c)

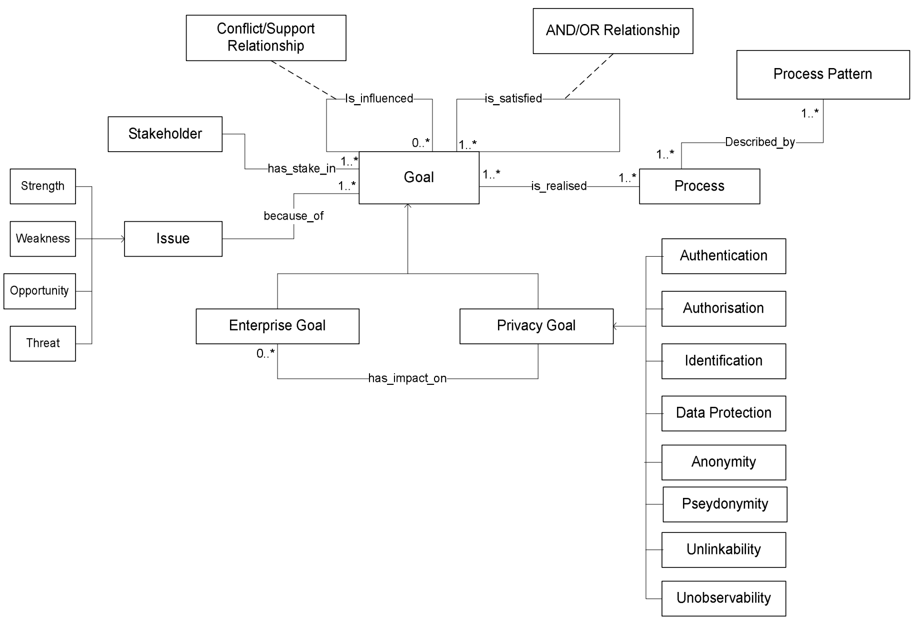


Fig. 1. PriS Conceptual Framework

the software systems that support the above processes. In this way, a connection between system purpose and system structure is established.

PriS models privacy requirements as a special type of goal (privacy goals) which constraint the causal transformation of organizational goals into processes. From a methodological perspective reasoning about privacy goals comprises of the following activities: (a) Elicit privacy-related goals, (b) Analyze the impact of privacy goals on business processes (c) Model affected processes using privacy process patterns and (d) Identify the technique(s) that best support/implement the above processes.

The first step concerns the elicitation of the privacy goals that are relevant to the specific organization. This task usually involves a number of stakeholders and decision makers who aim to identify the basic privacy concerns and interpret the general privacy requirements with respect to the specific application context into consideration. In addition, existing privacy requirements already forming part of the organization's goals are identified. The second step consists of two stages. In the first stage the impact of privacy goals on the organizational goals is identified and analyzed. In the second stage, the impact of the privacy goals on the relevant processes that realize these goals is examined and the processes that realize the privacy-related goals are identified and characterized as privacy-related processes. Having identified the privacy-related processes the next step is to model them, based on the relevant privacy process patterns. Business process patterns are usually generalized process models, which include activities and flows connecting them, presenting how a business should be run in a specific domain [4]. The last step is to define the system architecture that best supports the privacy-related process identified in the previous step. Once again, process pattern are used to identify the proper implementation technique(s) that best support/implement corresponding processes.

PriS assists in the application of privacy requirements in the organizational context as well as in providing a systematic way of locating a number of system architectures that can realize these requirements. PriS way of working assumes that privacy goals are generic-strategic organizational goals thus being mentioned high in the goal model hierarchy.

A formal expression of PriS can be found in [5]. A software tool for supporting PriS way of working has also been constructed and a detailed description can be found in [6].

3 The e-Voting case

PriS method is demonstrated through an e-voting case study, regarding the transformation of an Internet based electronic voting system in order to accommodate the new legal framework regarding privacy protection. The specific case study has been used for evaluation of previous versions of PriS as well. However, we consider the same case study in this paper as well in order to be able to test and validate the progress and effectiveness of our method by applying the proposed soft computing approach in comparison to our previously suggested versions of PriS.

The initial design of the electronic voting system was developed in the context of the European Project "E-Vote" by the University of Regensburg, in cooperation with the University of the Aegean, the Cryptomatic company, the Quality and Reliability company and the Athens University of Economics and Business and is described in [7]. According to this description, the main objective of the e-voting system is to provide eligible citizens the right to cast a vote over the Internet rather than visiting an election district, aiming to simplify the election processes thus increasing the degree of citizens' participation during elections. It is described by four main principles that form the four primary organizational goals namely: a) Generality, b) Equality, c) Freedom and d) Directness. Generality implies that all citizens above a certain age should have the right to participate in the election process. Equality signifies that both

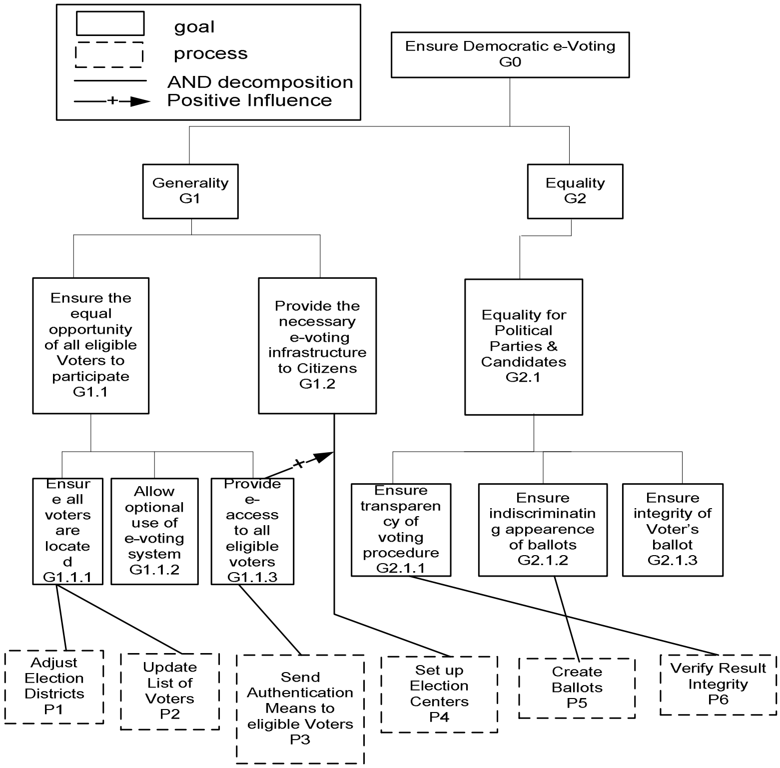


Fig. 2. Partial View of the e-Voting System Goal-Model

political parties - that participate in the election process - and voters have equal rights before; during and after the election process and neither the system nor any other third party is able to alternate this issue. Freedom implies that the entire election process is conducted without any violence, coercion, pressure, manipulative interference or other influences, exercised either by the state or by one or more individuals. Finally, directness means that no intermediaries chime in the voting procedure and that each and every ballot is directly recorded and counted.

A partial view of the system's current goal model is presented in Figure 2. In the last line the doted boxes are the relevant processes that satisfy organizational goals.

As mentioned earlier, the system has to be re-designed in order to guarantee that user's privacy is not violated. To this end, PriS was applied by two teams of postgraduate students of the University of the Aegean that worked in parallel in order to:

(a) to analyze the impact of privacy issues on the system's goals and processes and propose alternative system implementations (first team)

(b) formally describe the above process and its deliverables (second team)

(c) provide feedback regarding both difficulties encountered and recommendations or incorporation into the PriS method (both teams)

The students were computer science graduates and had knowledge of requirements engineering principles but no experience with the particular method. Work from this case study is reported in [4, 8]. The findings of this case study were cross checked with the ones of a second case study regarding the University of the Aegean Career Office System [9] which was conducted by two similar groups during the same period.

In the following section the application of the extended version of PriS (including the proposed fuzzy extension) is presented.

4 Applying Fuzzy PriS

In this section the PriS method is applied according to the four basic steps mentioned before. Through this case study the application of the new extension of PriS is also presented. Our main goal is to prove that the knowledge combination from a soft computing approach improves the way that PriS selects the respective PETs for the realization of the privacy requirements incorporated during the design phase, thus overcoming the drawback between design and implementation phases in a more robust and constructive way.

4.1 Elicitation of Privacy Related Goals

The first step concerns the elicitation of the privacy goals that are relevant to the specific organization. This task usually involves a number of stakeholders and decision makers (managers, policy makers, system developers, system users, etc). Identifying privacy concerns is guided by the eight privacy goal types shown in Figure 2. The aim is to interpret the general privacy requirements with respect to the specific application context into consideration. In the e-voting case two privacy goals were identified, namely: unlinkability and unobservability. The former refers to the voters' right to receive the respective authentication means (username and password) without others being able to reveal to whom the data are sent. Thus, even when a malicious third party is able to steal these data he/she won't be able to know neither the user nor the system where these data can be used. The latter concerns the voters' right to ensure the transparency of the e-voting procedure by verifying the results' integrity without other parties (either system users or malicious third parties which do not belong to the system) being able to observe the whole verification process.

It should be noted, that PriS assumes the existence of the organization's current goal model. If not, a goal modeling method should be used for constructing the goal model prior to PriS's application [10].

4.2 Analyze the Impact of Privacy Goals on Organizational Processes

The second step is to analyze the impact of privacy goals on processes and related support systems.

To answer this question, the first task is to identify the impact it may have on other organizational goals. This impact may lead to the introduction of new goals or to the improvement / adaptation of existing goals. Introduction of new goals may lead to the introduction of new processes while improvement / adaptation of goals may lead to the adaptation of associated processes accordingly. Repeating this process for every privacy goal and its associated organizational goals leads to the identification of alternative ways for resolving privacy requirements. The result of this process modeled in the spirit of an extended AND/OR goal hierarchy [8].

Let us consider the privacy goal of unlinkability in the e-voting case. Guaranteeing voters' unlinkability will clearly impact the way that goal '$G_{1.1}$: *Ensure the participation of all eligible voters'* is realized. In particular, by applying unlinkability goal on $G_{1.1}$, this will have an impact on all subgoals that realize goal $G_{1.1}$. For every subgoal it is analyzed which are the modifications that need to be done in order to satisfy the unlinkability goal. In the specific example, subgoals '$G_{1.1.1}$: *Ensure all Voters are located'* and '$G_{1.1.2}$: *Update List of Voters'* are maintained while goal '$G_{1.1.3}$: *Provide e-access to all eligible Voters'* needs to be adapted. Specifically, two new subgoals are introduced namely '$G_{1.1.3.1}$: *Provide e-access*' and '$G_{1.1.3.2}$: *Prevent others to reveal to whom the data are sent*' as the result of the impact analysis. Finally, the process that realizes these new subgoals is also adapted for accomplishing the realization of the new privacy goal [8]. The result of this analysis is graphically illustrated in Figure 3.

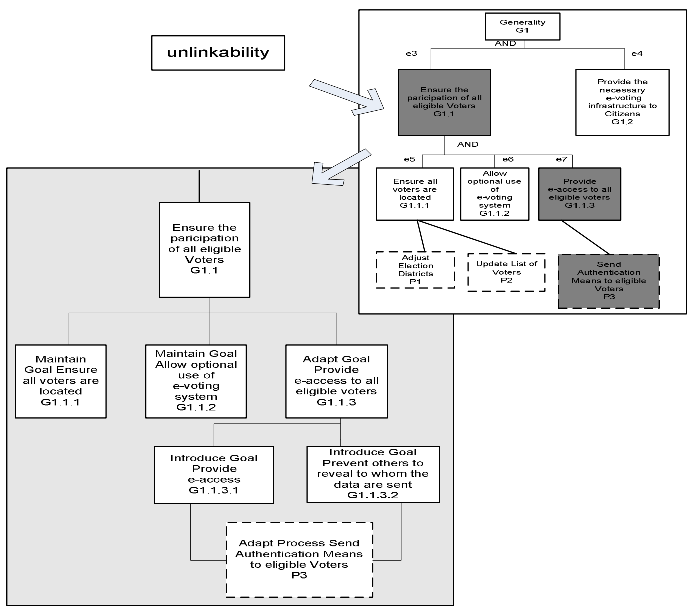


Fig. 3. Analyze the impact of unlinkability goal

4.3 Model Affected Processes Using Privacy Process Patterns

Having identified the privacy-related processes the next step is to model them, based on the relevant privacy process patterns. A detailed description of the seven privacy process patterns can be found in [8,11].

Figure 4 presents the process pattern for addressing the unlinkability requirement, which describes the relevant activities needed to realize that process. The application of the unlinkability pattern on process 'P3: *Send Authentication Means to eligible voters*', which realizes goals $G_{1.1.3.1}$ and $G_{1.1.3.2}$ as shown in Figure 4, is presented next to the general pattern.

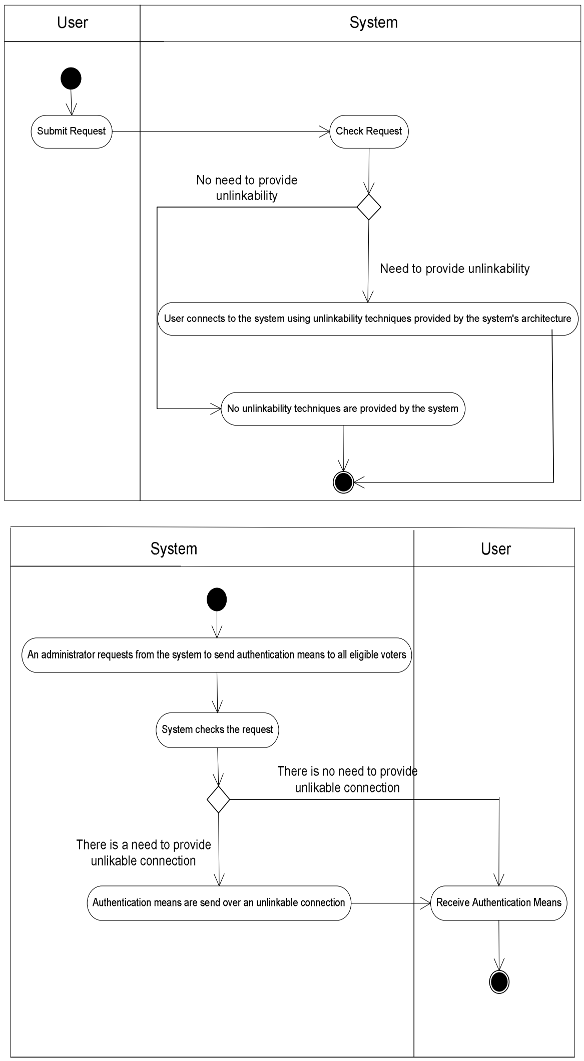


Fig. 4. Unlinkability Pattern & its Specialization on the e-voting case

4.4 Identify the Technique(s) that Best Support/Implement the Above Processes

For assisting the process of selecting the proper PETs for the realization of the respective privacy processes a table that matches the process patterns with a number of PETs is constructed and presented in a Table 1.

Table 1. Matching privacy patterns with implementation techniques

	Administrative Tools					Information Tools			Anonymizer Products, Services and Architectures											Pseudonymizer Tools		Track and Evidence Erasers				Encryption Tools		
	Identity Management	Biometrics	Smart Cards	Permission Management	Monitoring and Audit tools	Privacy Policy Generators	Privacy Policy Readers	Privacy Compliance Scanning	Browsing Pseudonyms	Virtual Email Addresses	Trusted Third Parties	Surrogate Keys	Crowds	Onion Routing	DC-Nets	Mix-Nets	Hordes	GAP	Tor	CRM Personalization	Application Data Management	Spyware Detection and Removal	Browser Cleaning Tools	Activity Traces eraser	Harddisk data eraser	Encrypting Email	Encrypting Transactions	Encrypting Documents
Authentication	X	X	X	X	X																							
Authorization	X	X	X	X	X																							
Identification	X	X	X	X	X																							
Data Protection	X	X	X	X	X	X	X	X																				X
Anonymity and/or pseudonymity	X	X	X	X					X	X	X		X	X	X	X	X	X	X	X	X	X			X			
Unlinkability											X	X		X		X	X	X	X	X	X	X	X	X	X			
Unobservability			X	X	X												X	X	X			X	X	X	X	X	X	X

Different tools in each category implement specific privacy process patterns. So far, using Table 1, a developer could choose for every process pattern which is/are the best implementation technique(s) among the ones available, always based on the privacy requirement(s) that needs to be realized, as well as the specific business context in which it will be implemented. However, this is not always the case since most organizations have a number of developers with different capabilities and opinions as well as various criteria with different weights that form the final decisions regarding which is the best technology that fits their organization.

Requirements engineering is a complex task that is affected by various factors. Among else, prioritization is a process that aims to determine which requirements should be given relative priority to the implementation process. When many project partners participate, often their personal experiences affect the way that they consider different requirements should be implemented first; in such a case, there is a need to establish a way to evaluate the different experiences and expertise and make a decision based on estimating all the participants' opinions. We describe methodological tools that provide a framework for decision support over conflicting evidence. We attempt to tackle with the issue of combining evidence from different experts in order to reach consensus in respect to the implementation of specific technique. The expert's opinion plays an important role in several parts of the development process; especially when decisions like the selection of tools for privacy and security come into focus. Instead of adopting a rigid process that demands consensus about the validity of a certain choice in respect to its evaluation for a number of factors, such as the cost or complexity, we prefer to utilize a method that lets the experts express on a scale their opinion about a specific choice and then combine the evidence from different sources.

Fuzzy theory provides solutions in the presence of vague or imprecise knowledge. Most of the cases people face decisions which cannot be made on a clear selection with a yes or no answer. Binary decisions are rare. In most of the cases, evidence that comes from different sources is hard to be managed. Fuzzy measures may be well suited in these cases; among else they provide a framework to treat variables that are spreading in the [0,1] interval. Evidence theory [12] is a branch of decision theory that utilizes fuzzy measures to handle uncertainty.

In evidence theory, and more specifically the branch which is acknowledfged as Dempster-Schafer of major importance are belief measures, which can be defined as a function mapping a given set to the [0,1] interval: Bel:P(X) → [0,1]. The belief measure may be interpreted as the degree of confidence that a fact is true or that a given element belongs to a set. It is obvious that if X is the set for which its subjects are considered, then the following relations stand: ($Bel(\varnothing)=0$, Bel(X)=1,

$$Bel(A_1 \cup A_2 \cup .. \cup A_n) \geq \sum_i Bel(A_i) - \sum_{i<k} Bel(A_i \cap A_k) + .. + (-1)^{n+1} Bel(A_i \cap A_2 \cap .. \cap A_n)$$

(1), for all subsets A of X.

Considering the facts A1, A2,.., An, are pair-wise disjoint, the inequality (1) requires that the belief required with the union of the sub-sets is no smaller than the sum of belief pertaining to each individual set.

The Belief metric can be represented by a function m: P(X) → [0,1], such that $m(\varnothing)=0$ and $\sum m(A)=1$. Function m(A) expresses the proportion to which available evidence supports the claim that a particular element belongs to A. In other words the relation between the metric and the supporting function can be expressed as: $Bel(A) = \sum_{B|B\subseteq A} m(B)$ (2).

The utility of the aforementioned measures is considerable in case that the evidence comes from independent sources for example from independent evaluators.

Therefore in order to calculate $m_{1,2}$ for the set A considering the evidence that focuses on subset B$\in$ P(X) and on the subset C$\in$ P(X) the following sum of products needs to be calculated: $\sum_{B\cap C=A} m_1(B)\bullet m_2(C)$ (3) for all $A \neq \varnothing$. Since $m_{1,2}(\varnothing)$ should equal to 0, we need to exclude the following sum of products of these subsets who's intersection results in the empty set: $\sum_{B\cap C=\varnothing} m_1(B)\bullet m_2(C)$. Since $\sum_{A\in P(X)} m(A)=1$, the combined evidence we are seeking is calculated if we subtract the value from 1 resulting in: 1-$\sum_{B\cap C=\varnothing} m_1(B)\bullet m_2(C)$. For normalization purposes the final result for the combined evidence $m_{1,2}(A)$ is given by the formula[14][16]:

$$m_{1,2} = \frac{\sum_{B\cap C=A} m_1(B)\bullet m_2(C)}{1-\sum_{B\cap C=\varnothing} m_1(B)\bullet m_2(C)} \quad (4).$$

Another issue worth noting is that by introducing, in our approach, the normalization factor $(1 - K)$ at the denominator we normalize the values and consider the appearance of strongly conflicting evidence as unlikely, associating thus such conflicts with the null set; in relevant literature there has been a lot of discussion on managing conflicts in evidence theory. In [13], Zadeh presents an example with a medical scenario in presence of conflicting evidence from different medical experts; it is shown that the combined evidence for unlikely events with high degrees of belief towards this unlikelihood, may result in a case where these not so probable events are given priority (due to the high support values on this unlikelihood). We need to clarify that in our examples, due to the nature of the software development process and due also to the fact that the variables have been specified from the beginning and are not assigned ad hoc by the evaluators, we do not consider that two different experts in the field will give conflicting evidence while examining the same parameters for the same technology.

Let's get back to the e-voting case. We consider independent experts who evaluate different solutions for a given privacy requirement. Considering that independent opinions may give different priorities to the existing requirements, we need to find an analysis tool that will enable us to reach a conclusion by incorporating these different evaluations. Traditional methods, and PriS so far, decide on a yes-no basis; this binary logic is hard to resolve conflicts in case when two different opinions lean differently, for example one is in favor of a specific technology while the other is less supportive. With traditional methods it would be hard to decide; fuzzy logic provides support to express intermediate opinions for example on the [0,1] interval.

Evidence theory [12][13][15] provides a methodological tool that considering the opinion of each member as well as an expressed support for this opinion, to make combined calculations and express the overall opinion of the group. In a given project we consider that a given set of requirements is achieved by implementing a given number of measures. As X we may consider the universal set of measures that implement a specific requirement. We consider next the subsets N, A and C that: a) the first subset N includes the measures that are by presumptive evidence essential in implementing a specific requirement, b) the set A includes the measures that are cost-efficient (affordable) and provide a value for money and c) is the set of measures that their complexity is such that allows their integration into a given software project. In our case study we had two different partner organizations that were responsible to select the appropriate technologies for the implementation of specific requirements.

In the specific case now we need to find the most appropriate technologies for realizing process P3 on which unlinkability process pattern is applied. According to table 1 a subset of candidate technologies (all available technologies are shown in the respective table) for realizing unlinkability in P3 is:

- Trusted Third Parties
- Onion Routing
- DC-Nets
- Mix Nets
- GAP
- Hordes
- Tor

The aim is to apply for the list of available solutions the ones that are considered by both parties as more appropriate, in terms of satisfaction of the given parameters that affect the decision:

a) the necessity of a measure,
b) the cost for its implementation, and
c) the complexity for its development.

We asked from the two parties to assign a value for each of the three parameters for two of the available solutions. Then we combine the evidence from the two sources according to equation (4) so that the outcome produces the combined evidence. The same process can be applied iteratively when more factors are considered for a given project. Initially, we examined the case Tor. The two parties assigned a value for the three parameters (necessary, affordable and complex, independently, as well as for their combination, for example necessary and affordable at the same time). The Bel metric shows the belief and is calculated using equation 2. Table 2 presents the values m_l assigned by the first partner, while column 3 represents the respective values from the second partner. The 2nd and 4th columns are calculated and finally we extract the combined values for the combined evidence $m_{1,2}$ and Belief: $Bel_{1,2}$. Fig. 5 shows the values from Table 2.

From what is apparent, both parties give more value to the necessity and cost of the solution, as they have assigned in general higher values to these two metrics. The combined value for N and A are considerably high, which means that both parties consider that this solution should be implemented and also that it would not cost enough. We also see that they still consider it as complex in respect to other potential solutions.

We also considered the case for Hordes presented in Table 3 and figure 6 respectively. This time we see that both parties grade very low the cost of the solution. This results to very low combined values for the parameter A. But the combined values give to the project managers a tool to estimate the trend from the participants.

Table 2. m_l assigned from the two developers regarding Tor

	m_1	Bel_1	m_2	Bel_2	$m_{1,2}$	$Bel_{1,2}$
N	0.03	0.03	0.15	0.15	0.20	0.20
A	0.03	0.03	0.22	0.22	0.22	0.22
C	0.02	0.02	0.06	0.06	0.08	0.08
$N \cup A$	0.2	0.26	0.07	0.44	0.12	0.54
$A \cup C$	0.2	0.25	0.12	0.4	0.16	0.46
$N \cup C$	0.1	0.15	0.05	0.26	0.07	0.35
$N \cup A \cup C$	0.42	1	0.33	1	0.15	1.00

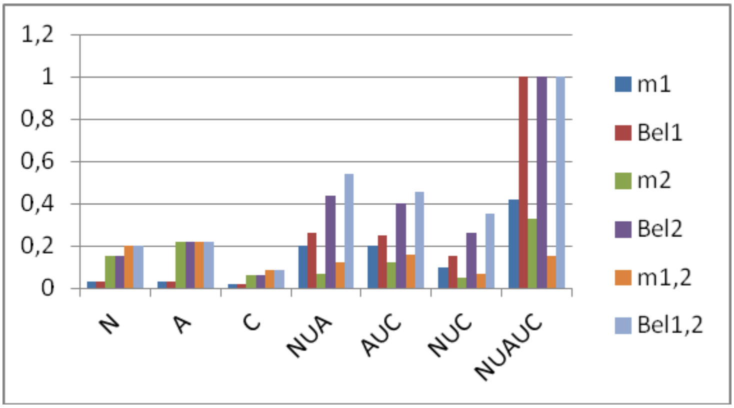


Fig. 5. Graphical representation of the values of Table 2

Table 3. m_1 assigned from the two developers regarding Hordes

	m_1	Bel_1	m_2	Bel_2	$M_{1,2}$	$Bel_{1,2}$
N	0.04	0.04	0.15	0.15	0.21	0.21
A	0.01	0.01	0	0	0.02	0.02
C	0.03	0.03	0.05	0.05	0.08	0.08
NUA	0.17	0.22	0.05	0.2	0.13	0.36
AUC	0.09	0.13	0.2	0.25	0.19	0.29
NUC	0.06	0.13	0.05	0.25	0.07	0.36
NUAUC	0.6	1	0.5	1	0.31	1.00

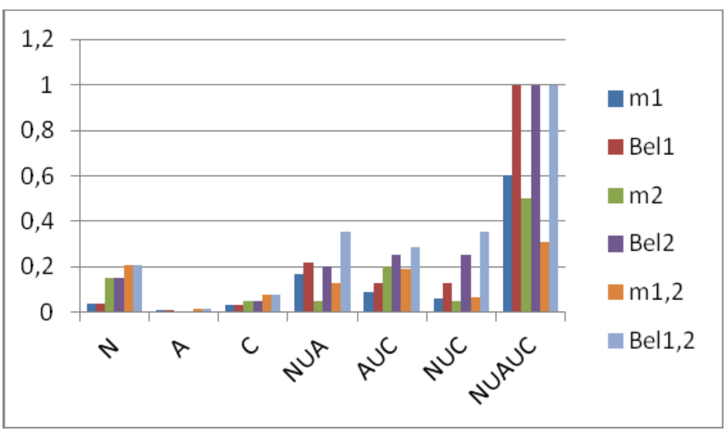


Fig. 6. Graphical representation of the values of Table 3

5 Conclusions

Decision making in software design process is not always straightforward; often the implementation of specific privacy related countermeasures depends on the evaluation of different factors for which often opinions vary among the project partners. In this paper an extension of PriS method is presented. On the main drawbacks of PriS was on the selection of proper PETs for the realization of the privacy process patterns identified during system design. While PriS bridges the gap between design and implementation it fails on the way of suggesting the implementation techniques. Specifically, the selection was conducted without any criteria rather than a yes/no criterion based on Table 1.

Thus, we have extended PriS by providing methodological tools that help the developers estimate the most appropriate solutions by considering combined opinions from independent sources while developing privacy measures in the software design process. The aforementioned method enables also to tackle a serious problem of estimating the combined opinions in a formal manner. It is important also to note that the method also is by no means limited by the number of independent evaluations nor by the number of subsets (factors) considered prior to making the decision.

References

[1] Kalloniatis, C., Kavakli, E., Gritzalis, S.: Methods for Designing Privacy Aware Information Systems: A review. In: Alexandris, N., Chryssikopoulos, V., Douligeris, C., Kanellopoulos, N. (eds.) Proceedings of the PCI 2009 13th Pan-Hellenic Conference on Informatics (with International Participation). IEEE CPS Conference Publishing Services, Corfu (2009)

[2] Loucopoulos, P., Kavakli, V.: Enterprise Knowledge Management and Conceptual Modelling. In: Chen, P.P., Akoka, J., Kangassalu, H., Thalheim, B. (eds.) Conceptual Modeling. LNCS, vol. 1565, pp. 123–143. Springer, Heidelberg (1999)

[3] Loucopoulos, P.: From Information Modelling to Enterprise Modelling. In: Information Systems Engineering: State of the Art and Research Themes, pp. 67–78. Springer, Berlin (2000)

[4] Kavakli, E., Gritzalis, S., Kalloniatis, C.: Protecting Privacy in System Design: The Electronic Voting Case. Transforming Government People Process and Policy 1(4), 307–332 (2007)

[5] Kalloniatis, C., Kavakli, E., Gritzalis, S.: Addressing privacy requirements in system design: The PriS method. Requirements Engineering 13(3), 241–255 (2008)

[6] 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); AIS SIGSEC

[7] University of the Aegean, E-Vote: An Internet-based electronic voting system. University of the Aegean, Project Deliverable D 7.6, IST Programme 2000#29518, Samos (November 21, 2003)

[8] 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)

[9] 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 (2005)
[10] Kavakli, V.: Goal Oriented Requirements Engineering: A Unifying Framework. Requirements Engineering Journal 6(4), 237–251 (2002)
[11] 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)
[12] Klir, G., Yuan, B.: Fuzzy sets and fuzzy logic. Prentice-Hall (1995)
[13] Zadeh, L.A.: Review of Books: A Mathematical Theory of Evidence. The AI Magazine 5(3), 81–83 (1984)
[14] Yager, R.: Quasi-Associative Operations in the Combination of Evidence. Kybernetes 16, 37–41 (1987)
[15] Sentz, K., Ferson, S.: Combination of Evidence in Dempster-Shafer Theory, Sandia National Laboratories, Technical Report SAND 2002-0835, Albuquerque, New Mexico (2002)
[16] Inagaki, T.: Interdependence between Safety-Control Policy and Multiple-Sensor, Schemes Via Dempster-Shafer Theory. IEEE Transactions on Reliability 40(2), 182–188 (1991)

A Meta-model for Legal Compliance and Trustworthiness of Information Systems

Fatemeh Zarrabi, Michalis Pavlidis, Haralambos Mouratidis, Shareeful Islam, and David Preston

School of Architecture, Computing and Engineering, University of East London
s.zarrabi@uel.ac.uk, m.pavlidis@ieee.org,
{haris,shareeful,david}@uel.ac.uk

Abstract. Information systems manage and hold a huge amount of important and critical information. For this reason, information systems must be trustworthy and should comply with relevant laws and regulations. Legal issues should be incorporated into the system development process and there should be a systematic and structured assessment of a system's trustworthiness to fulfil relevant legal obligations. This paper presents a novel meta-model, which combines legal and trust related concepts, to enable information systems developers to model and reason about the trustworthiness of a system in terms of its law compliance. A case study is used to demonstrate the applicability and benefits of the proposed meta-model.

Keywords: Hohfeld taxonomy, natural language pattern, legal constraint, trustworthy information systems, trust modelling, control.

1 Introduction

Information systems in the modern world exist in every aspect of human life. Governmental organizations, factories, and hospitals (to name few) deploy such systems to manage huge amount of sensitive and critical information. As such, security of such information systems is of paramount importance. Any security failure of those systems can cause potential losses of money, time, or even life. In such cases liability of information systems is assessed and information system owner should be constituted responsible to replace damages [1]. To this end, information systems should comply with relevant laws. However, information system developers face two main challenges. Firstly, developers need to capture requirements from legal texts, align them with other system requirements and assign them to relevant system components. Secondly, to ensure law compliance, system developers need to place trust and rely on human actors and software components. The trustworthiness of such actors and components to achieve their legal duties needs to be assessed properly during the development of a system. Otherwise, if these actors and components are not trustworthy, they can possibly harm the ability of a system to fulfil legal obligations and be law compliant. However, the current literature fails to support

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 46–60, 2012.

information systems developers with adequate practices and methods to face those challenges. On one hand, although there is some work to support capturing of requirements from legal texts (see section 5 for more information), the literature fails to provide clear evidence of appropriate frameworks and methodologies to support the capture and analysis of system requirements from relevant laws and regulations. On the other hand, there is lack of frameworks to support the analysis of trustworthiness, within the context of law and regulation compliance, at the requirements engineering stage.

This paper presents a first step towards the development of a novel framework that overcomes the above problems by combining concepts from trust engineering and regulatory requirements capturing. In particular, the paper presents a meta-model that enables developers to model legal requirements during information systems development and reason about the trustworthiness of the actors (human or software components) who are assigned and responsible for the fulfilment of those requirements. The proposed meta-model is based on legal concepts such as legal constraints, duties and rights [2] that are assigned to actors, and trust-related concepts, such as experiential, reported, normative, and external trust, and control [3] over the enforcement of duties.

The paper is structured as follows. In section 2 we describe relevant legal and trust related concepts. The meta-model is presented and discussed in section 3, while in section 4 we demonstrate the applicability and benefits of the proposed meta-model with the aid of a case study. In section 5 we discuss related work and section 6 concludes the paper.

2 Legal and Trust Concepts

Laws, regulations and policies related to an information system need to be incorporated into the system development process since they constitute restrictions on the system. Laws and regulations use very high-level language and technical terminologies, which represent legal instructions within obligations and recommendations. Being unfamiliar with technical language of laws has made considerable challenge for system developers to understand laws and indeed to identify the stakeholders and the action of law. To overcome this problem, current work is taking advantage of a legal taxonomy called Hohfeld [4] along with the aid of some natural language patterns. Hohfeld analyses laws by separating them into two main groups of legal relations between individuals. The first group indicates legal respected choice of individuals and is called *Right.* Based on Hohfeld, *Right* is paired with a correlative called *Duty*. Duty is the second category of legal relationship introduced by Hohfeld and indicates which one ought or ought not to do. Therefore, one person's right for an action against another entails the *duty* of the other against the first person in respect of that action and vice versa. Right and Duty, as the main two groups of legal relations, also contain four subgroups that inherit correlative relations from their parent group (Table 1).

Table 1. Hohfeld Legal Rights and correlative Duties

Right	Claim	Power	Immunity	Liberty
Duty	Duty	Liability	Disability	No-Right

Claim is entitlement of a person to have something done from another person and it correlates to duty. For example a contract between employer and employee confers on the employee the right to be paid his wages, which he/she can claim for this right and it correlates to the duty of employer to pay the wages to employee. Liberty is one's freedom from the right or claim of another and it is paired to the correlative of no-right. Suppose that people are free and have the right to smoke in an open environment, therefore no one has a right to prohibit them from smoking. Also, power is one's affirmative control over a given legal relation as against another. For example, a librarian has the power over a student with regard to the use of the library. Normally the student has the right to use the library, but if he is noisy the librarian has the power to take away that right and stop the student from using the library. Immunity is one's freedom from the legal power or control of another. For example, diplomats are supposed to have diplomatic immunity, which means that if they commit a crime in the hosting country, they are immune against arrest and legal prosecution. In other words, the hosting country police are disabled to act [4]. In legal documents, duties and rights of stakeholders are expressed using specific modality notations, for instance 'shall', 'must', 'may' (Table 2).

Table 2. Modality Notations

Duty	Shall, must, shall be, must be, shall prohibit, may not, shall not, must not, is required to
Right	May, may be, shall guarantee the right, has the right to

As seen in Table 2, modality notations are divided in two different categories based on the extent of control they enforce on the action. To identify modality notations, our approach employs a language pattern called Modality Pattern [5].

The phrase "Language Pattern" refers to specific samples related to linguistic typology used in language grammars in order to identify different units in a sentence (Table 3). To identify other elements of law such as subject, verb and object we use basic activity pattern. Whereas our basic activity pattern is following SVO (Subject-Verb-Object) sentence structure where the subjects come first, verb second, and object third [6]. Therefore the process of using these patterns can be used for most of legal languages. The difficulty of this activity to identify these elements from a sentence is related on the complexity of the sentence itself. Sometimes a sentence simply consists of basic activity pattern elements (subject, action, object) or the object itself consists of one or more other sentences. Noun phrases such as "who", "which" and others give more details on the above mentioned elements and are used to identify scope of law or to extract extra requirements from law. Conditional, exceptional and purpose patterns also extract extra requirements of system since they restrict the action of law. They are mandatory requirements of system since they are enforcing some specific circumstances on duties and rights.

Table 3. Natural Language Patterns

Language Pattern	Text's element	Meaning	Identifying phrases
Basic activity pattern	Subject	Who performs the action	-
Basic activity pattern	Object	What the action is performed on	-
Basic activity pattern	Action	What is performed	-
Noun phrase pattern	Target	Action is performed on whom	To, on, of, from, …(It can also be identified from the concept of object or subject)
Purpose pattern	Purpose (goal)	Why the action is performed	To, in order to, …
Condition pattern	Condition	When and in which condition action should be performed	If, when, whenever, …
Exception pattern	Exception	When and in which situations action should not be performed	Except when, except that, is except from, …
Modality pattern	Modality phrases	If the action is required, or recommended to be performed	Must, shall, may, may not, must not, is required to, has a right to, …

In order to perform compliance of modelled laws on the desired system context, we need to map extracted elements from laws to that system context. For this purpose, we adopt the i* modelling language [7]. I* models social relationships between stakeholders of a system environment using the concepts of actor, dependency, goal, plan and resource. In i* actors depend on each other to achieve a goal, carry out a plan or deliver a resource. These concepts are useful in order to represent the relationships between the different stakeholders. However, we extend i* with law related concepts in order to represent the legal relationships, and with trust related concepts in order to reason about trust. The rest of this section provides a brief description of the adopted i* concepts [8].

Actor. Actor is an entity of the domain of interest that possesses strategic goals and can carry out actions that will fulfil those goals. It is a unit that encapsulates intentionality, rationality, and autonomy.

Goal. A goal is a condition that an actor wants to achieve. Of course, this can be done in more than one way so alternatives of achieving a goal can be considered.

Plan. A plan is a procedure that has to be followed in order to accomplish a goal and specifies the way of achieving that.

Resource. A resource is an informational or a physical entity that is needed to accomplish a goal or to carry out a plan.

Dependency. A Dependency is a relationship between two actors. In this relationship one actor is the depender and the other actor is the dependee. The depender is depending on the dependee to satisfy the dependum, which is the object around which the relationship centres. The dependum can be a goal, a plan or a resource.

3 Law and Trust Meta-model

Figure 1 shows the meta-model that combines both law and trust related concepts. The concepts with orange colour are the law related concepts [2] and these concepts are linked with the trust concepts [3] with the grey colour.

Secure Goal. Secure goal is the strategic interest of an actor with respect to security.

Legal Constraint. Legal constraint is the restriction related to legal issues such as when an actor is required and instructed to comply with a law in order to achieve a goal, perform a task or receive a resource.

Duty. Duty is something that one is expected or required to do by legal obligation. It covers Hohfeld's obligations of duty, liability, disability and no-right since they emphasise on actions that are required to be done by actors and are not optional. A Duty *correlates* to a Right of target party of law having a mutual and complementary relationship and based on Hohfeld taxonomy who believes duty of a stakeholder cannot exist without right of another party. A Duty will be satisfied with a goal or secure goal using Mandatory satisfy link. This will enforce the necessity of the fulfilment of the goal or secure goal since they are ordered by law and cannot be refused or ignored.

Right. Right is something that one is allowed or recommended to do or owned by law. It covers Hohfeld rights of claim, power, immunity and liberty since they represent optional decision of actors to perform an action or not. Right of a stakeholder also correlates to Duty of other party based on Hohfeld taxonomy. A Right can be satisfied by goals and secure goals taken by the correlated duty-bearer or it may be satisfied by goals and secure goals which are needed to be taken by the right-holder himself. This is due to the reason that a right may always not be satisfied by a duty.

Resolution. Resolution is the indication of how the uncertainty in the fulfillment of a duty dependency is removed in order to build confidence in that duty dependency. There are two types of resolution, i.e., trust and control, which the developer uses to feel confident about the dependency.

Trust. Trust is the positive expectation of one actor about the behaviour of another actor by whom he might be positively or negatively affected [9]. The actor who trusts another actor is called trustor and the actor that is being trusted is called trustee. Trust can be decomposed into four types. They are:

- **Experiential Trust.** Experiential trust is trust that originates from previous direct experience with the trustee. The depender then is actually depending on himself.
- **Reported Trust.** Reported trust is trust that originates from a third party (the reporter) who reports that the trustee is trustworthy. Therefore, depender depends on the reporter to trust the dependee. As a result, reported trust creates an indirect trust relationship with the reporter. This new indirect trust

relationship is required in order to support the direct trust relationship with the actor that has duties.

- **Normative Trust.** Normative trust is trust that originates from the system environment norms. The depender is then depending on the system norms.
- **External Trust.** External trust is trust that originates from sources outside of the system environment. The depender is the depending on the external source of trust.

Trust Relationship. Trust relationship is defined as a relationship that exists between the trustor and the trustee and resolves a dependency based on trust. There can be direct and indirect trust relationships. Direct trust relationships are the trust relationships with actors that are responsible for fulfilling duties, while indirect trust relationships are trust relationships with actors that exist in order to justify the direct trust relationships or control relationships.

Control. Control is the power of one actor to enforce the fulfilment of a duty by another actor. This eventually means that the actor has the ability to gather information about another actor and also the ability to influence the other actor's present and future. In other words, it is the ability to influence the other actor's goals. When the type of resolution is control then there is a third party who acts as a controller. In particular, when there are legal constraints, actors are required to possess duties in order to satisfy these legal constraints. In this case the dependency on actors to achieve the duties has a control type of resolution that means that a third party acts as a controller. So, there is a new indirect trust relationship with the controller that supports the control relationship.

Entailment. Entailment is a trust assumption and needs to be examined if it is true or not. These assumptions need evidence in order to be justified. For example, if there is an entailment that a certain employee is trusted, is there any evidence that supports this assumption? If the outcome is positive then there is confidence that the actors will fulfil their duties. The system then satisfies the legal constraints and it is trustworthy in terms of law compliance Otherwise, there are possible vulnerabilities that can lead to legal breaches.

The initial step of using the meta-model is to consider the correlative theory of Hohfeld to find all relevant relations of stakeholders. Based on correlative theory of rights, when a stakeholder of a law has a right, the opposite party has the duty against him and vice versa. Our main consideration is on the function of rights and duties. In other words, on what the rights and duties bring to their holders from the requirements point of view. We achieve this goal by correlating a Right with a Duty and satisfying the duty with goals and secure goals. Also a Right can be satisfied by a goal or secure goal since there are cases that rights cannot be satisfied with goals taken by correlative duty-bearer and they need to be satisfied with goals that right-holder takes. We are also representing Duty and Right relationships with the aid of Dependency concept from i*. Therefore right-holder and duty-holder depend on each other or on other actors in order to perform a duty or claim for a right. The reason behind the use of the concept of Dependency is that in a system environment an actor can never work

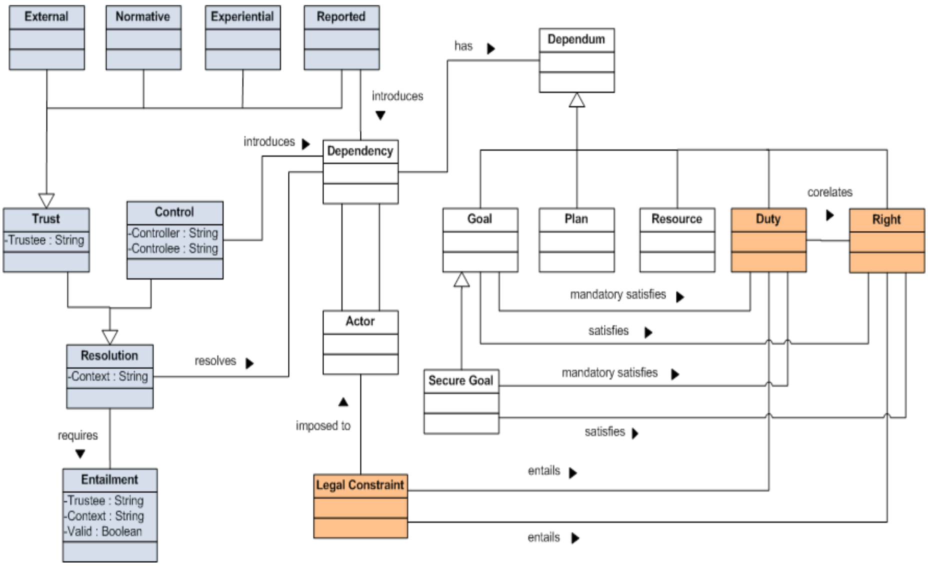


Fig. 1. Law and trust meta-model

without a dependency on other actors and there is always cooperation and relationship between two or more actors, for each actor to satisfy his goals. Also, we cannot ignore the consideration of legal rights in the design of a system since this is important to emphasize on the mandatory of the legal rights. Therefore, the system developer should be precisely instructed on actions that he/she is required to do (duty) and has the option to do (right).

Nevertheless, there is no guarantee that the actors assigned with duties will fulfil them. Further analysis of whether such actors are trusted to fulfil their duties is required in order the developer to be able to make informed and justified decisions during the development process. Verifying that actors are indeed trusted will remove any uncertainty and build confidence that the developed system will be trustworthy and law compliant.

As said before, an actor wants to achieve goals, carry out plans and deliver resources. However, there are legal requirements, which are represented as legal constraints, and are imposed to an actor in order to restrict the achievement of one or more of the actor's goals. To satisfy the legal constraints, the actor needs to achieve duties. At this point the trust-based concepts are introduced in order to be able to reason if actors are trusted to achieve duties, in order to ensure the law compliance of the system. There is an uncertainty whether the actors are willing to achieve duties, so trust and control are used to resolve the dependencies on such actors. The dependencies on actors to achieve duties require a resolution that specifies how the confidence is built. The resolution can be through trust and/or control. Trust and control reduce the uncertainty and increase the confidence in the actors. Nevertheless, the resolution of dependencies on actors with duties create entailments, which are conditions of trust that need to be validated in order to have confidence that actors will achieve their duties that will ensure the law compliance of the system.

4 Case Study

For the purpose of demonstrating the usefulness of our approach we will use a case study based on Dropbox. Dropbox [10] provides file-hosting services to internet users over the web. The files are stored and shared in cloud storage and particularly in Amazon's S3 storage system. Also, Dropbox collects and stores the files that are being uploaded by the users, information about the device used, its software, and user activity. Dropbox is required to comply with DMCA (Digital Millennium Copyright Act of United States) [11], which establishes a notification-and-takedown system for addressing claims of copyright infringement. Analyzing the system environment of the Dropbox application there is a number of different actors involved. These are:

- Dropbox Engineer. Engineer is responsible for maintenance of Dropbox service and the datacenter.
- User. The internet user of the Dropbox service.
- Amazon S3. The cloud service that Dropbox is using to store files.
- Legal Team. Responsible for the legal issues of Dropbox.
- Dropbox System. It is the technical system of Dropbox service.

Based on the system context of this scenario, we only consider the actors relevant to Dropbox uploading system. Also, size of this paper does not allow the practice of all related laws and duties and rights. There is a legal constraint that Dropbox technical system needs to comply with Digital Millennium Copyright Act (DMCA). Therefore there is number of duties and rights assigned to the Dropbox system in order to comply with this Act which are extracted from the following clauses.

1) DMCA.SEC.103 INTEGRITY OF COPYRIGHT MANAGEMENT INFORMATION

(a) FALSE COPYRIGHT MANAGEMENT INFORMATION. —No person shall knowingly and with the intent to induce, enable, facilitate, or conceal infringement distribute or import for distribution copyright management information that is false.

2) DMCA.SEC.202. INFORMATION STORED ON SERVICE PROVIDERS.

(1).A service provider shall not be liable for monetary relief, or except as provided in subsection (i) For injunctive or other equitable relief, for infringement for the storage at the direction of a user of material that resides on a system or network controlled or operated by or for the service provider, if the service provider

A) Does not have actual knowledge that the material or activity is infringing,

B) Service provider responds expeditiously to remove or disable access to the materials claimed to be infringing upon notification to claimed infringement.

Table 4. Extracted text elements from DMCA. SEC103. (a)

Element	Extracted item	Language pattern
Subject	Person	Basic Activity
Action	Distribute, Import for distribution	Basic Activity
Object	False copyright management information	Basic Activity
Target	Copyright-holder	Using concept of subject and object
Scope on action	Knowingly and with the intent to induce, enable, facilitate, or conceal infringement	Noun Phrase Pattern
Modality Notation	Shall not	Modality Pattern

Table 5. Extracted elements from DCMA. SEC.202. (1)

Elements	Extracted item	Language Pattern
Subject	Legal authority	Using concept of subject and object
Action	Make liable (implicate)	Basic Activity
Scope on action	For monetary relief, or except as provided in subsection for injunctive or other equitable	Noun Phrase Pattern
Target	Service provider	Basic Activity
Object	Storage of materials	Basic Activity
Scope on object	Materials: that resides on a system	Noun Phrase Pattern
Scope on storage	Storage is made at the direction of user	Noun Phrase Pattern
Scope on system or network	Controlled or operated by or for the service provider	Noun Phrase Pattern
Mandatory notation	Shall not	Modality Pattern
Condition	1-Without knowledge of infringing 2-Remove or disable access to materials	Conditional Pattern
Scope on condition2	Upon obtaining such knowledge or awareness	Noun Phrase Pattern
Scope on condition3	In the instance of a notification of claimed infringement as described in paragraph	Noun Phrase Pattern

Table 4 and 5 show the items that are extracted from DMCA. SEC103.(a) and SEC202.(1) using mentioned Language patterns in last column. We have used these extracted elements together with Hohfeld concepts of rights to identify Right and Duty dependencies and their correlatives between subject and target as stakeholders of law, as they are mentioned in Table 6.

Type of dependency is identified based on Modality Notation if it indicates duty or right. Using categorization of right (Table 1) extracted right in Sec103 is a type of claim-right. Therefore the copyright-holder has the right to claim if he believes false copyright management information of him had been distributed. This claim makes a requirement in Dropbox system, for infringement notifications and claims and also to have access to its users shared files to check the validity of the claim. This requirement and a list of other requirements extracted from duties and rights are satisfied through concept of goal and are mentioned in Table 7. Each of mentioned goals should be satisfied by the dependency between a dependee and depender as mentioned in Table 7.

Table 6. Duty & Right dependencies extracted from DCMA. SEC103 & SEC202

DCMA Section	Law's actor	Dropbox actor	Target party	Duty or Right Dependency
Sec. 103	Person	User (Dependee)	Copyright-holder Dropbox system (Depender)	DD1: duty not to distribute or import for distribution false copyright management information
Sec. 103	Copyright-holder	Copyright-holder (Dependee)	Dropbox system (Depender)	RD1: right that false copyright management information of him/her don't be distributed or imported for distribution
Sec. 202	Legal Authority	Legal Authority	Dropbox system (Depender)	DD2: duty not to implicate Dropbox for monetary relief, or except as provided in subsection for injunctive or other equitable relief on storage of materials in mentioned conditions
Sec 202	System provider	Dropbox system (Dependee)	Legal Authority (Depender)	RD2: the right not to be liable for monetary relief, or except as provided in subsection for injunctive or other equitable relief on storage of materials in mentioned conditions

To analyze mentioned clause in Sec202, the first step is again to extract the exact duty from the text using extracted elements in Table 5and correlate it with the right of law's target stakeholder based on Hohfeld taxonomy (Table 6). For example Dropbox system has the right not to be liable for monetary relief of any copyright infringement if it did not have knowledge of infringement or has deleted false materials in case of knowledge. (Table 6. RD2). Since we have a strict condition in this text, duty dependency2 (Table 6. DD2) contains a mandatory goal2, which enforces service provider to remove or disable access in case on knowledge on its infringement of copyright. Also, this goal entails other goals since in order to be able to remove or disable access the service provider should have access to saved materials and also the authority to cancel an account. Also, from the extracted scopes on each of the main elements of the text (Table 5) such as scope on object and storage, we were able to extract other goals such as save the materials, control or operate a service or a goal to notify the service provider. RD2 from Table 6 also entails some other requirements. For example, in order to have knowledge of incident, service provider need to have a notification facility which this requirement is extracted from previous article. Service provider also needs to keep notifications for a period of time to prove his authority to remove materials or cancel user's account. Service provider is able to remove materials or cancel user's account. In order to satisfy this requirement, service provider needs to have access to user's account, which is another requirement. This requirement is not clearly mentioned in the law text but is extracted from molecular analysis of the duty of supervisor.

Since all mentioned requirements are extracted from legal resource, therefore there is a mandatory of their existence in the designed system regardless if they are extracted from rights or duties. In case of right, the requirement and its satisfactory goal, plan and resource should be available in system, but the related actor may decide to claim and use her/his right or not. Therefore, we are expressing the mandatory of extracted requirements from legal text with the aid of a mandatory satisfy link which can ends to a goal, plan or resource and is followed till the termination of system design.

Table 7. Extracted requirements from Duty & Right dependencies

Duty & Right Dependencies	Depender	Dependee	Goals
RD1	Copyright-holder	Legal	• Notify about infringement
DD2	Legal Authority	Dropbox System	• Check copyright-holder notification • Check system log histories
RD2	Dropbox System	Dropbox Engineer	• Disable user access • Remove materials • Access to materials • Disable account
RD2	Dropbox System	Net Engineer	• Control a service or network • Save system log histories for period of time • Keep copyright-holder notifications for a period of time

There is an uncertainty though if actors are trusted to fulfill their duties. If they are not trusted then there should be some form of control on them in order the developer to feel confident that the duties will be fulfilled. In our case study actors have a number of goals that are required to be fulfilled as part of the fulfillment of their duties. Therefore, the developer needs to resolve the dependencies on the actors with duties. The resolutions of the duty dependency on the Dropbox Engineer are shown in table 8.

Table 8. Resolutions

Duty Dependencies	Resolutions
Dropbox Engineer disables user access	1. Normative Trust 2. Legal Team controls the Dropbox Engineer
Dropbox Engineer removes materials	3. Normative Trust 4. Legal Team controls the Dropbox Engineer
Dropbox Engineer accesses materials	5. Normative Trust 6. Legal Team controls the Dropbox Engineer
Dropbox Engineer disables account	7. Normative Trust 8. Legal Team controls the Dropbox Engineer

The resolutions of duties that are based on control though create new duty dependencies on the Legal Team to control the Dropbox Engineer to fulfil various duties. These new duty dependencies need to be resolved again in order the developer to feel confident that the Legal Team will fulfil its duties. The new resolutions are shown in table 9.

Table 9. New resolutions

Duty Dependencies	Resolutions
Legal Team controls Dropbox engineer to disable user access	9. Normative Trust
Legal Team controls Dropbox engineer to remove material	10. Normative Trust
Legal Team controls Dropbox engineer to access material	11. Normative Trust
Legal Team controls Dropbox engineer to disable user account	12. Normative Trust

The identified resolutions create entailments, which are conditions of trust that are required to be valid in order the analysis that has been carried so far to be based on correct trust assumptions. The required entailments and the resolutions from which they originate are shown in table 10.

Table 10. Entailments

Resolution	Entailments
1	System norm is trusted for Dropbox engineer to disable user access
2	Legal Team is trusted to control Dropbox Engineer to disable user access
3	System norm is trusted for Dropbox Engineer to remove material
4	Legal Team is trusted to control Dropbox Engineer to remove material
5	System Norm is trusted for Dropbox Engineer to access material
6	Legal Team is trusted to control Dropbox Engineer to access material
7	System Norm is trusted for Dropbox Engineer to disable user account
8	Legal Team is trusted to control Dropbox engineer to disable user account
9, 10, 11, 12,	System norms are trusted for Legal Team to control Dropbox Engineer for the respective goals

The above entailments were validated with evidence that was collected from the case study. Dropbox is a small size company where all employees are located in the same office. Also, the Legal Team and the Dropbox Engineer have close collaboration and this enables the Legal Team to control the Dropbox Engineer. Therefore, since the entailments are valid then there is confidence in the fulfilment of duty dependences. Thus, the analysis of legal issues was based on trust assumptions that are valid and as a result the Dropbox System can be trusted to be law compliant.

5 Related Work

Number of researches have analysed laws and regulations in order to extract right and obligations of law's stakeholders. Breaux et al. [12] has used natural language patterns in order to elicit rights and obligations from laws and regulation. He has used HIPAA law (Health Insurance Portability and Accountability Act) to extract security requirements of health systems. Later he extends the work by to analyse access control rules to it. Islam in [5] also has used natural language patterns with Hohfeld legal taxonomy to extract security requirements from laws and combine them with the ISO/IEC policies and has integrated extracted constraints into UMLsec for secure detail design of a system. May at [13] have extracted privacy requirements from legal text using access control techniques. Dorimont et al. [14] have modelled regulations using GORE software modelling methodology (Goal Oriented Requirement Engineering). The special GORE approach that he has used is KAOS which starts modelling from goal and refine goals in an incremental process through leading to relevant tasks and involved actors. Siena et al. at [15] has only focused on Hohfeld legal taxonomy to extract security and privacy requirements from laws and regulations. Mead et al. [16] has introduced a method called SQUARE (Security Quality Requirement), which elicits and documents security requirements. Mellado et al. [17] has presented a security requirement engineering process based on Common Criteria and ISO/IEC 270001, which can be used as a constant method to develop system based on these two policy references.

Among mentioned works, most have tried to elicit security requirements from laws. Some have only concentrated on special laws such as HIPAA and a case study based on that law. To demonstrate a framework, we need to analyse more number of most applicable laws in order to have a more valid framework that works in different cases. Some of mentioned works only elicit security requirements without considering laws. The advantage of this work is to align legal requirements with other requirements of a system, first to answer to enforcement of compliance with extending current state of the art of software development methodologies with legal concepts; second to elicit more requirements of system considering laws and molecular analysis of rights and duties. Among the above mentioned works some also have introduced legal concepts during development of software system, but the advantage of this work compared to theirs is using of an actor-goal oriented software methodology which has the capacity to map legal relationship of stakeholders to its own language using concept of dependencies between actors of system. The reason of this usage is that we believe an only goal-oriented or actor-oriented methodology lack the contiguous concepts of laws and software development and consequently makes the process difficult.

There are a number of approaches that consider trust issues, including trust modelling and the formation of a common vocabulary during the software development stage. In [18] the proposed method makes use of the Goal Requirement Language (GRL) and Use Case Map (UCM) which both of them belong to the User Requirement Notation (URN). Specifically, trust is captured as a soft goal because of the uncertainty of whether it has been satisfied or not and because of its fuzzy nature. Further analysis of trust as a soft goal eventually leads to well-defined tasks. Yu and Liu [19] address the issues of trust at the requirements level of the system development process. They consider trust as a non-functional requirement, where trust is a combination of all or some quality attributes of a system under development and they demonstrate their approach by describing the behaviour of a system in the case of attack and examine defences that are needed from trust perspective. Secure Tropos [20] extends Tropos methodology with the concepts of trust, delegation, provisioning and ownership in order to allow the developer to capture trust relationships at a social and individual level. Bimrah [21] extends the Secure Tropos [22] methodology with the concepts of request, action, trust relationship, trusting intention, reputative knowledge, recommendation and consequence in order to model trust. The developer is guided through a series of models in order to analyse and reason about trust relationships.

However, the above-mentioned approaches, and in particular, [18] treat the system as a black box without looking into the trust relationships inside the system, thus concentrating on the trust relationships between user and the system. On the other hand, in the cases such as [19], [20], and [21] where trust relationships are modelled they are not justified or they are limited to the direct trust relationships omitting the indirect ones that can become a serious vulnerability to the trustworthiness of the system. We believe our work contributes in this direction by providing a meta-model that supports the capture and reasoning of the direct and indirect trust relationships inside the socio-technical system and identifying the gaps in the chain of trust relationships. With the use of the proposed meta-model, there is the advantage that these trust relationships become explicit and the developer can reason about them in order to develop confidence in them. Otherwise, if any trust relationship is left unidentified it could become a potential vulnerability to the functionality and proper operation of the final system. The main contribution though of the meta-model presented in this paper is that not only allows the elicitation of requirements from laws and regulations but more importantly it allows the trust analysis of the actors that are related with the law compliance of the system.

6 Conclusion

In this paper we have presented a meta-model that combines legal and trust related concepts. It enables the developer to model the legal issues that introduce legal requirements to the system design and constitutes the developed system law compliant. Also, the meta-model enables the assessment of trustworthiness of the actors that are assigned and responsible to fulfil legal duties. The incorporation and analysis of the legal issues is explicitly carried out in order to show its importance and the trustworthiness of the actors involved in the fulfilment of the legal obligation is assessed in a structured and coherent way.

In addition, the applicability and benefits of the meta-model were demonstrated by using a scenario from the popular Dropbox service. Legal requirements were identified and the trustworthiness of the actors responsible for legal duties was assessed in a systematic way in order to ensure that the system is trustworthy in terms of law compliance.

Acknowledgments. The second author would like to acknowledge funding from the Engineering and Physical Sciences Research Council (EPSRC) and British Telecom (BT).

References

1. Ryan, J.D.: Two Views on Security Software Liability: Let the legal System Decide. In: Mead, R.N., McGraw, G. (eds.) IEEE Security & Privacy, pp. 70–72. IEEE Computer Society Press (2003)
2. Zarrabi, F., Mouratidis, H., Islam, S.: Extracting Security Requirements from Relevant Laws and Regulations. In: Proceedings of the International Conference on Research Challenges in Information Science (2012)

3. Pavlidis, M., Mouratidis, H., Islam, K.P.: Dealing with Trust and Control: A Meta-Model for Trustworthy Information Systems Development. In: Proceedings of the International Conference on Research Challenges in Information Science (2012)
4. Hohfeld, W.N.: Fundamental Legal Conceptions as Applied in Judicial Reasoning. Yale Law Journal 23(1) (1913)
5. Islam, S., Mouratidis, H., Jürjens, J.: A Framework to Support Alignment of Secure Software Engineering with Legal Regulations. Journal of Software and Systems Modeling (SoSyM), Theme Section on Non-Functional System Properties in Domain-Specific Modeling Languages (NFPinDSML) 10(3), 369–394 (2011)
6. The Cambridge Encyclopaedia of Language. Cambridge University Press (1997) ISBN 0-521-55967-7
7. Yu, E.: Towards Modelling and Reasoning Support for Early-Requirements Engineering. In: Proceedings of the 3rd IEEE International Symposium on Requirements Engineering, pp. 226–235 (1997)
8. Yu, E., Liu, L., Mylopoulos, J.: A Social Ontology for Integrating Security and Software Engineering. In: Mouratidis, H., Giorgini, P. (eds.) Integrating Security and Software Engineering: Advances and Future Visions, pp. 70–105. Idea Group Publishing, London (2007)
9. Mollering, G.: The Trust/Control Duality: An Integrative Perspective on Positive Expectations of Others. International Sociology 20(3), 283–305 (2005)
10. Dropbox, `http://www.dropbox.com`
11. House of Representatives: Conference Report: Digital Millennium Copyright Act. Report 105-796 (October 1998)
12. Breaux, T.D., Antón, A.I.: Analyzing Regulator Rules for privacy and Security Requirements. IEEE Transactions on Software Engineering 34(1) (January-February 2008)
13. May, M.J., Gunter, C.A., Lee, I.: Privacy APIs: Access Control Techniques to Analyze and Verify Legal Privacy Policies. In: Proc. of the 19th Computer Security Foundations Workshop (July 2006)
14. Darimont, R., Lemoine, M.: Goal-oriented Analysis of Regulations. Regulations Modeling and their Validation and Verification (2006)
15. Siena, A., Mylopoulos, J., Perini, A., Susi, A.: From Laws to Requirements. In: 1st International Workshop on Requirements Engineering and Law (2008)
16. Mead, N.R.: Identifying Security Requirements Using the Security Quality Requirements Engineering (SQUARE) Method. In: Integrating Security and Software Engineering, pp. 44–69. Idea Publishing Group (2006)
17. Mellado, D., Medina, E., Piattini, M.: A common criterion based security requirements engineering process for the development of secure information system. Computer Standards & Interfaces 29, 244–253 (2007)
18. Pourshahid, A., Tran, T.: Modelling Trust in E-Commerce: An Approach Based on User Requirement. In: Proceedings of the 9th International Conference on Electronic Commerce, USA, pp. 413–422 (2007)
19. Yu, E., Liu, L.: Modelling Trust for System Design Using the *i** Strategic Actors Framework. In: Falcone, R., Singh, M., Tan, Y.-H. (eds.) AA-WS 2000. LNCS (LNAI), vol. 2246, pp. 175–194. Springer, Heidelberg (2001)
20. Giorgini, P., Massaci, F., Mylopoulos, J., Zanone, N.: Requirements Engineering for Trust Management. International Journal of Information Security 5(4), 257–274 (2004)
21. Bimrah, K.K.: A Framework for Modelling Trust during Information Systems Development. PhD Thesis, University of East London (2009)
22. 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)

Ontologies for Security Requirements: A Literature Survey and Classification

Amina Souag[1], Camille Salinesi[1], and Isabelle Comyn-Wattiau[2]

[1] CRI , Sorbonne University Paris 1, France
{Amina.Souag,Camille.Salinesi}@malix.univ-paris1.fr
[2] CEDRIC-CNAM & ESSEC Business School, France
isabelle.wattiau@cnam.fr

Abstract. Despite existing methodologies in the field, most requirements engineers are poorly trained to define security requirements. This is due to a considerable lack of security knowledge. Some security ontologies have been proposed, but a gap still exists between the two fields of security requirement engineering and ontologies. This paper is a survey, it proposes an analysis and a typology of existing security ontologies and their use for requirements definition.

Keywords: Security, Ontologies, Requirements, Analysis, Classification.

1 Introduction

Security of Information Systems (IS) has progressively become a very broad research field [1]. Security is defined as a discipline which allows one to build reliable systems that can face malice, errors or mischief [2]. The domain of IS security also encompasses a set of methods, techniques and tools responsible for protecting the resources of an IS to ensure information availability, confidentiality, integrity, and traceability. A requirement prescribes a property judged necessary for the system; security requirements engineering frameworks derive security requirements using security-specific concepts, borrowed from security engineering paradigm. With the growing need to implement IT security measures in world-wide corporate environments and the growing application scope, a major obstacle, that face ordinary analysts and developers using existing security requirements modeling and analyzing frameworks, is the lack of security knowledge and expertise [3][29]. Ontologies are useful for representing and interrelating many types of knowledge [4],[5]. In 2003, Marc Donner argued that too much security terminology is vaguely defined, thus it becomes difficult to communicate between colleagues and, worse, confusing to deal with the people we try to serve [6]. Since that, many security ontologies have been proposed during the last decade. But there are still questions around these works: what are the different security ontologies available nowadays? Do they meet the requirements? Do they cover all or some security aspects? Which ontology can I choose as an analyst seeking for security knowledge for the definition of IS requirements? We faced these questions, and concluded that we definitely need a

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 61–69, 2012.

general survey of existing security ontologies. Analysts and researchers may find in this paper a road map, an overview of what exists in terms of security ontologies. Our main objective is to review, analyze, select, and classify security ontologies, as a scope study but with a particular interest in the field of security requirements engineering. The rest of the paper is organized as follows: in Section 2 we explain the methodology used in the study. Section 3 includes the survey and classification, and Section 4 recalls related works. Finally, Section 5, the conclusion, raises future perspectives.

2 Methodology of Research

To perform this survey, we relied on information retrieval and survey methodologies presented in [7,8]. We started by gathering any publication related to ontologies, requirements, security and its various aspects. The search was conducted inside the relevant and known sources of literature such as ACM libraries, IEEE digital library, etc. About 50 papers were gathered. We performed a first read to get a general idea; 21 papers were discarded at this stage when they were found to be far away from our target objective. A second read was carried out for deeper understanding and analysis of concepts and relations between them. Finally, a qualitative analysis lead us to classify them into different families, and we defined a set of criteria allowing us to compare the approaches. The result of this comparison is synthesized in Table 1.

3 Synthesis and Classification

The framework of our classification is composed of 8 families of security ontologies (Fig. 1), described as follows:

- **Beginning security ontologies:**
One of the earliest work (back in the nineties) about merging knowledge base and information system management at an early level of development was [9] which proposed a language (Telos) and a knowledge base divided into four sub-worlds. Mylopoulos et al. note that Telos users can develop models for the purpose of security specification.
- **Security taxonomies:**
Taxonomies of security concepts are a common method for sharing security knowledge. Avizienis et al. [10] provide a detailed taxonomy that contains classes of *faults*, *fault modes*, *fault tolerance techniques*, and *verification* approaches. McDermott et al. [11] were particularly interested in security flaws.
- **General security ontologies:**
By general ontologies we mean these ontologies which aim at covering all (or most) security aspects; Herzog and colleagues [12] endeavored to deliver an extensible ontology that includes both general concepts and specific vocabulary of the domain. In the same vein, Fenz and Ekelhart [13] have proposed an ontology that has a similar goal but attempts to cover non-core concepts such as the infrastructure of organizations.

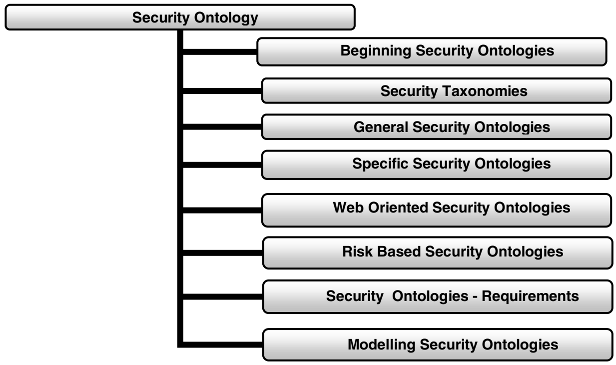


Fig. 1. Classification of Security Ontologies into 8 families.

- **Specific security ontologies:**

This category gathers the security ontologies dedicated to a specific domain. In [14], the authors propose a data model that characterizes the domain of computer attacks and intrusions as an ontology that covers concepts like (*Host, System Component Attack, input, Consequence).* Geneiatakis and Lambrinoudakis [15] propose an ontology for SIP-VoIP based services. This ontology can be applied either to find a countermeasure against attacks on SIP based VoIP services or for testing the security robustness of SIP-VoIP (Session Initial Protocol-VoIP) infrastructures.

- **Web oriented security ontologies**:

Denker et al in [16,17,18] develop several ontologies for security annotations of agents and web services, using DAML and later OWL. The defined ontology is composed of two sub-ontologies: "*security mechanisms*" and "*credential*". The NRL Security Ontology proposed in [19] is organised around seven separate ontologies. Artem Vorobiev and Jun Han proposed a security attack ontology for Web services [20].

- **Risk based security ontologies**:

Recent trends in security methodologies tend to consider that the best approach of security consists in starting from a risk analysis. Fenz et al. [3] proposed a security ontology framework based on four parts (the security and dependability taxonomy from [10], the underlying risk analysis methodology, the concepts of the IT infrastructure domain, and a simulation enabling enterprises to analyze various policy scenarios). Lenne et al. [21] proposed to develop a knowledge base containing ontologies for the analysis of industrial risks describing concepts used for the achievement of a risk analysis.

- **(Security) Ontologies for Security requirements:**

Some papers refer to ontologies in order to cope with the definition of security requirements: Dobson and Sawyer [5] propose an ontology of dependability by merging two conceptualisation models (IFIP model & UMD model). Tsoumas et al. [22] define a security ontology using *Asset*, *Stakeholder*, *Vulnerability*, *Countermeasure* and *Threat* concepts. In [23] (Karyda et al.), the authors proposed an ontology formed of "*assets*" (data asset, hardware data, etc.), "*countermeasures*" (identification and authentication, network management, auditing services, physical protection, etc.), "*objectives*", "*persons*" and "*threats*" (errors, attacks, technical failures, etc.). Firesmith [24] presents a taxonomy of security-related requirements (pure security requirements, security-significant requirements, security system requirements, security constraints).

- **Security modelling ontologies:**

Even if authors present them as ontologies, they mainly describe metamodels. While the previous ontologies include security specific concepts such as threat, attack, vulnerability, these ontologies include security related concepts for modelling requirements and the dependencies between them such as relationship, proposition, situation. Thus Mouratidis et al. [25] and Massacci et al. [26], proposed ontologies respectively associated with Tropos and i*.

4 Discussion and Evaluation

In this section, we present the result of our comparison and evaluation of the ontologies by using a set of criteria (security objectives, assets, vulnerabilities, threats, countermeasures and organisation). *Security objective* defines which security objective (accountability, confidentiality, integrity) are wished to be reached and can be affected by a certain threat. A *threat* is anything (manmade or act of nature) that has the potential to cause harm, it exploits a vulnerability. A *vulnerability* is considered as a property of the system or environment which, in conjunction with an attack, can lead to a failure. An *asset* is defined as something valuable in an *organization*. *Assets* are subject to attacks. The *countermeasures* are sets of actions implemented in prevention of the threat [27].

Mylopoulos in [9] did not literally propose a security ontology, but a basic taxonomy composed of four sub-worlds. The authors note that users of Telos have developed models for the purpose of security specification but did not detail the underlying models. Avizienis et al. [10] fail to cover techniques for protecting confidentiality and establishing authenticity. The taxonomy was not used for requirements definition. The main limit in the taxonomy of McDermott et al. [11] is that it is too basic, focused on some flaws in operating systems only, far from many kinds of security flaws that might occur in application programs for database management, electronic mail, and so on. The two general security ontologies of Herzog et al.[12] and Fenz et al. [13] are both interesting contributions but neither of them is complete. While the first one seems simple and clearer, the second is much richer but more complex. Fenz et al. cover better asset concepts, while Herzog et al. are better focused on threat concepts. Fenz's main contribution consist of the organisation concepts, clearly absent from Herzog. Herzog's countermeasures tend to be technical whereas Fenz's are both business and technical. The advantage of these ontologies of being generic and capturing most security criteria leads also to drawbacks since they lack in specificity that the domain dedicated security ontologies provide, and vice-versa. Neither [12] nor [13] ontologies were used for requirements definition and analysis. The general ontologies offer generic concepts of security objectives, assets, vulnerabilities, countermeasures, threats, etc. while the rest offers more specific threats concepts (computer attacks and intrusions in [14], for example). The web oriented security ontologies do not cover some aspects like vulnerabilities or threats. Nevertheless, Kim et al. [19] proposed a matching algorithm that facilitates mapping of higher level (mission-level) security requirements to lower-level (resource level) capabilities using the ontology. In a very similar previous work by Denker et al. [16,17,18], the proposed ontology fails to consider vulnerabilities, assets and threats; but a reasoning engine matches between the request requirements and the capabilities of a potential web service. The risk based security ontologies of Fenz et al. [3] and Lenne et al. [21] could be useful for a risk based requirement analysis. However, to the best of our knowledge, there are no propositions combining both sides. In the context of requirements engineering some ontologies were proposed, but unfortunately none of them is associated to a methodology describing how to use them for requirement definition. Dobson and Sawyer's ontology [5] concentrates on few threat concepts. Tsoumas et al. [22] don't indicate any detailed mechanism on how to use the ontology for requirement collection. Finally, the security modelling ontologies, which are more security modelling oriented might be useful for constructing security requirements models like Secure i*.

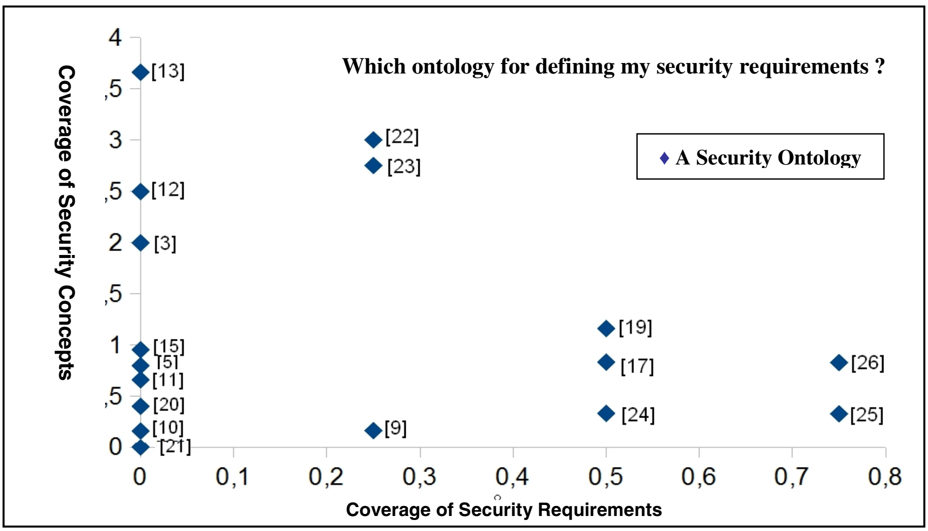


Fig. 2. Using security ontologies for requirement definition

We summarise this analysis and evaluation in Table 1. The rows are the security ontologies. The columns are the security concepts (criteria of the study: objectives, assets, vulnerabilities, threats, countermeasures and organisation). The last column in the table evaluates the link between the ontology and requirements definition. A black dot measures to which extent the security ontology covers this specific criterion, and how this particular security ontology deals with requirements. We used a dash for

Table 1. Summary of security ontologies of the study

[1] How does the ontology cover this concept of security? [2] How does this security ontology deal with requirements (last column)?

Family	Security ontology	Security Objectives[1]	Assets[1]	Vulnerabilities[1]	Threats[1]	Counter-measures[1]	Organisation[1]	Requirements[2]
Beginning	Mylopoulos et al. [9]	-	-	-	-	-	●	●
Security Taxonomies	Avizienis et al. [10]	●●●	-	-	●●	●●●●	-	-
	McDermott et al. [11]	-	-	-	●●●●■	-	-	-
General	Herzog et al. [12]	●●	●●●■	●●	●●●■	●●●■	●●	-
	Fenz et al. (a)[13]	●●●●	●●	●●●●	●●●●	●●●●	●●●●	-
Specific	Undercoffer et al. [14]	-	●■	●■	●●■	-	-	-
	Geneiatakis et al. [15]	●●	-	-	●●●■	-	-	-
Risk based	Fenz et al. (b)[3]	●●	-	-	●●●● ■	●●●	●●●	-
Web oriented	Denker et al. [16] [17] [18]	●●●	-	-	-	●●■	-	●●
	Kim et al. [19]	●●●●	-	-	-	●●●■	-	●●
	Han et al. [20]	-	-	-	●●●●■	-	-	-
For security requirements	Dobson et al.[5]	●●	-	-	●●	-	-	-
	Tsoumas et al.[22]	-	●●●	●●●	●●	●●●	●	●
	Karyda et al. [23]	●●	●●●	-	●●●	●	●●	●
	Firesmith [24]	-	●	-	●	-	-	●●●
Modelling	Mouratidis et al.[25]	●	-	-	-	-	●	●●●
	Massacci et al. [26]	●●	●●	-	●	-	-	●●●

- : absent ●: very few ●●: few ●●●: much ●●●●: very much Does this ontology refer to technical security concepts? ■: Technical absence of use and a black square to indicate that technical aspects of security were addressed, as follows: (-: absent ●: very few ●●: few ●●●: much ●●●●: very much ■: Technical). To complete the study we drew up a graph that represent roughly, for each security ontology, how much it deals with requirements (x-axis) and how much it covers security concepts (y-axis). The graph in Figure 2 clearly reveals a gap between the two fields. There is not a perfect ontology that covers lots of security aspects and, at the same time, that can be used in the definition for security requirements.

5 Related Works

While many security ontologies have been proposed, few surveys have been attempted. The only ones we can cite here are [1], [27], and recently [28] who proposed a survey of general ontologies for information systems encompassing some security ontologies. Blanco et al. [1] contains an interesting review and comparison of security ontologies that helped us in our study. However, since 2009 other ontologies have been proposed, indicating a need for updating. Moreover, Blanco et al. organized the existing ontologies under four categories. Our aim was to extend this classification and to update their surveys with recent literature contributions.

6 Conclusion and Perspectives

This study has shown the existence of considerable work around security ontologies; We classified the proposed ontologies into eight families. This classification extends the previous works which were limited to two, three, or four families at best. Our analysis has also shown that the existing security ontologies vary a lot in the way they cover security aspects. We tried to analyse how each ontology covers each aspect of security which formed our criteria of the analysis. Moreover, we studied whether the proposed security ontology can be used for requirements definition and the degree of this use. The study revealed a real gap between the fields of security requirement engineering and ontologies, and thus a new area of research to explore. We believe that this work can be improved; the classification needs to be extended. We also believe that there are still important issues to be addressed in the adaptation of ontology-based requirements engineering techniques to security requirements Engineering. This paper allows us to assert that the challenges facing software security is the lack of an easily accessible large common body of security knowledge. It remains difficult for designers to extract relevant pieces of knowledge to apply to their specific design or requirements related decision making situations. Our objective for the next steps of our research is to explore the best use of these security ontologies for security requirement definition.

References

[1] Blanco, C., Lasheras, J., Valencia-Garcia, R., Fernandez Medina, E., Toval, A., Piattini, M.: A systematic review and comparison of security ontologies. In: ARES, Barcelona (2008)
[2] Mouratidis, H., Giorgini, P., Manson, G.: Towards the development of secure information systems: Security Reference Diagram and Security Attack Scenarios. In: CAiSE (2004)
[3] Ekelhart, A., Fenz, S., Klemen, M., Weippl, E.: Security Ontologies: Improving Quantitative Risk Analysis. In: HICSS 2007 (2007)
[4] Gruber, T.R.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal Human-Computer Studies 43(5-6), 907–928 (1995)
[5] Dobson, G., Pete, S.: Revisiting Ontology-Based Requirements Engineering in the age of the Semantic Web. Requirements Engineering (2006)
[6] Donner, M.: Toward a Security Ontology. IEEE Security and Privacy (2003)
[7] Barnes, S.J.: Assessing the value of IS journals. Communications of the ACM (2005)
[8] Rainer, R.K., Miller, M.D.: Examining differences across journal rankings. Communications of the ACM 48(2), 91–94 (2005)
[9] Mylopoulos, J., Jarke, M., Koubarakis, M.: Telos – a language for representing knowledge about information systems. ACM Trans. Information Systems 8(4), 327–362 (1990)
[10] Avizienis, A., Laprie, J.-C., Randell, B., Landwehr, C.E.: Basic concepts and taxonomy of dependable and secure computing. IEEE Trans. Dependable Sec. Comput. (2004)
[11] Landwehr, C.E., Bull, A.R., McDermott, J.P., Choi, W.S.: A taxonomy of computer program security flaws. ACM Comput. Surv. 26(3), 211–254 (1994)
[12] Herzog, A., Shahmehri, N., Duma, C.: An Ontology of Information Security. International Journal of Information Security 1(4), 1–23 (2007)
[13] Fenz, S., Ekelhart, A.: Formalizing information security knowledge. In: ASIACCS 2009, pp. 183–194 (2009)
[14] Undercoffer, J., Joshi, A., Pinkston, J.: Modeling Computer Attacks: An Ontology for Intrusion Detection. In: Vigna, G., Krügel, C., Jonsson, E. (eds.) RAID 2003. LNCS, vol. 2820, pp. 113–135. Springer, Heidelberg (2003)
[15] Geneiatakis, D., Lambrinoudakis, C.: An ontology description for SIP security flaws. Computer Communications (2006) (in press) (corrected proof)
[16] Denker, G., Kagal, L., Finin, T.W., Paolucci, M., Sycara, K.: Security for DAML Web Services: Annotation and Matchmaking. In: Fensel, D., Sycara, K., Mylopoulos, J. (eds.) ISWC 2003. LNCS, vol. 2870, pp. 335–350. Springer, Heidelberg (2003)
[17] Denker, G., Nguyen, S., Ton, A.: OWL-S Semantics of Security Web Services: a
[18] Denker, G., Kagal, L., Finin, T.: Security in the Semantic Web using OWL. Information Security Technical Report 10(1), 51–58 (2005)
[19] Kim, A., Luo, J., Kang, M.: Security Ontology for Annotating Resources. In: ODBASE 2005 (2005)
[20] Vorobiev, A., Han, J.: Security Attack Ontology for Web Services. In: SKG 2006, p. 42. IEEE Computer Society (2006)
[21] Abou Assali, A., Lenne, D., Debray, B.: Ontology development for industrial risk analysis. In: ICTTA 2008, Damascus, Syria (April 2008)
[22] Tsoumas, B., Gritzalis, D.: Towards an ontology-based security management. In: AINA, pp. 985–992 (2006)
[23] Karyda, M., et al.: An ontology for secure e-government applications. In: ARES 2006, pp. 1033–1037. IEEE Computer Society (2006)

[24] Firesmith, D.G.: A taxonomy of security-related requirements. In: RHAS 2005, Paris (2005)
[25] Mouratidis, H., Giorgini, P., Manson, G.: An Ontology for Modeling Security: The Tropos Approach. In: Knowledge-Based Intelligent Information and Engineering Systems, pp. 1387–1394. Springer, Heidelberg (2003)
[26] Massacci, F., Mylopoulos, J., Paci, F., Tun, T., Yu, Y.: An extended ontology for security requirements. In: WEISSE 2011, June 20-24 (2011)
[27] Case Study. In: 1st European Semantic Web Symposium: Heraklion, Greece (2004)
[28] Elahi, G.: Security Requirements Engineering: State of the Art and Practice and Challenges (2009), `http://www.cs.utoronto.ca/~gelahi/DepthPaper.pdf`
[29] Nguyen, V.: Ontologies and Information Systems: A Literature Survey (2011), `http://hdl.handle.net/1947/10144`
[30] Sikora, E., Tenbergen, B., Pohl, K.: Industry needs and research directions in requirements engineering for embedded systems. In: RE (2012)

A Model for Time-Awareness

Francesco Fiamberti, Daniela Micucci,
Alessandro Morniroli, and Francesco Tisato

University of Milano - Bicocca, Viale Sarca 336, Milan, Italy
{fiamberti,micucci,tisato}@disco.unimib.it,
alessandro.morniroli@gmail.com

Abstract. Time-aware activities are characterized by a set of time-related aspects, independently from the involved application domain. For example, an activity may need to reason on facts that are held to be true in specific time intervals, or it may need to be executed at precise time instants. In this paper we present a temporal model capturing these concepts and their relations. The model is described by means of an UML formalization, enriched with OCL constraints where needed. The model turns into a set of architectural abstractions that makes time-related concepts visible at the application level. This eases the analysis and implementation of time-aware systems and enables adaptivity so that temporal constraints may be dynamically met.

Keywords: time-aware activities, time, timeline, clock, UML.

1 Introduction

It is widely recognized that a sound software architecture based on a proper set of *architectural abstractions* ([1] and [2]) can bridge the gap between requirements and specifications [3]. Architectural abstractions are design abstractions representing architectural aspects of a system. Well-known examples of architectural abstractions are the *architectural styles* as detailed in [4].

Domain-specific architectural abstractions capture aspects that are relevant in a specific domain so that they can be treated at the architectural level. Under this perspective, time-aware systems [5] are systems that have to deal with *time-related* issues when accomplishing domain-related tasks. For example, a time-aware system may include activities whose activation is *time-driven*, activities that need to reason on *time-stamped facts*, and activities whose *execution period* must be controllable. Therefore, such kind of systems claim for a set of suitable domain-specific architectural abstractions that allows both the temporal behavior of the system to be represented and time-related concepts to be treated as first class entities at the application level. The abstractions should rely on a sound temporal model aimed at specifying all the significative concepts (including their characteristics and relations) that allow a time-aware system to reason on time-stamped information, to know what time it is, and to observe and control the execution period of its time-driven activities.

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 70–84, 2012.

Different research communities agree that time ought to emerge as a first-class concept because of its relevance in the application domain. From the ontologies' research community point of view, the temporal dimension is central in many information sources. Unfortunately, languages like OWL (Ontology Web Language) [6] and RDF (Resource Description Framework) [7] provide a limited support for enriching data with temporal information. As stated in [8], there are many drawbacks in lacking such a support: a minor expressivity of the temporal aspect of the data, a difficult automatization of the validation process and a reduced temporal expressivity of deductive rules and queries on data. As a result, ontologies often cannot fully express the required temporal knowledge, forcing the adoption of ad-hoc solutions.

Different proposals have been conceived to represent temporal aspects in both OWL and RDF. An up-to-date detailed survey may be found in [8]. Dealing with RDF, many extensions to the model have been proposed so that temporal information can be represented (e.g., [9]). According to such extensions, several query languages have been conceived so that new queries are made available exploiting temporal extensions (e.g., [10]). As stated in [11], temporal description logics are not able to represent temporally-changing information, since they are geared towards synchronic relationships, not diachronic ones. This is one of the reasons why researchers do not try to extend the OWL model, but try to propose extensions on the top that allow temporal information to be modeled. In this direction, an interesting result is Time-OWL ([12] and [13]), an ontology of temporal concepts aimed at describing the temporal content of Web pages and the temporal properties of Web services. The ontology includes topological properties of instants and intervals, measures of duration and the meaning of clock and calendar terms. Even if several researchers have extended the OWL description logic model (e.g., [14] for a survey) by adding temporal modeling, there not exists an agreement of a standard. Moreover, few research has been conducted regarding query languages supporting such extensions. A recent step towards the integration of temporal information and related query support is [8], that proposes a methodology and a set of tools for representing and querying temporal information in OWL ontologies.

From a software engineering perspective, the focus is on various aspects related to the construction of real-time system, ranging from modeling tools to schedulability analysis tools up to concrete frameworks.

As stated in [15], only recently model-based development has begun focusing on timing aspects of a system in addition to its functional and structural ones. However, the proposed approaches tend to specify the requirements with respect to timing focusing on a specific solution. Moreover, the satisfaction of timing requirements is verified only during the test phase of the development process. As stated in [16], this is principally due to the fact that model-driven approaches applied to embedded systems in early design phases do not always rely on a systematic and rigorous methodology that includes specifying and verifying timing requirements.

MARTE (Modelling and Analysis of Real-Time and Embedded systems) [17] and its predecessor SPTP (Profile for Schedulability, Performance, and Time) [18], are UML profiles designed to face real-time aspects of a system from a model-based perspective. A UML profile customizes UML for a specific purpose or domain by using extension mechanisms able to modify the semantics of the meta-model elements [19]. MARTE models time-related concepts with specific elements like clocks and provides two models that respectively allow to describe the execution platform and the allocation of the functional components to the resources. AADL [20] is an analysis and design language that allows not only to define a representation of the software architecture, but also to formally define the syntax and semantics so that the representation can be verified and validated [21]. Both the approaches, however, only provide modeling capabilities but no embedded tools to directly implement the system. Languages like Giotto [22] and SIGNAL [23] extend existing paradigms to include time-related issues. However, such issues are managed at compile time, preventing the temporal behavior of the system from being adaptive. Similar to Giotto, PTIDES (Programming Temporally Integrated Distributed Embedded Systems) [16] is a programming model for distributed embedded systems based on a global, consistent notion of time. Finally, [24] proposes a modular modeling methodology to specify the timing behavior of real-time distributed component-based applications. It allows building models of the resources and of the software components, which are reusable and independent from the applications that use them.

The key idea behind our proposal is that time-related concepts should be full-edged first-class concepts, which directly turn into basic architectural abstractions supported by a running machine. In this way, it is possible to explicitly treat time-related aspects from the analysis of the requirements to the test phase of the life cycle of a system. More important is that, thanks to abstractions, time emerges at the application level. Therefore, it is possible to build adaptive systems that dynamically change their behavior by observing timed facts, the current time and the execution speed of the activities. Particularly interesting is that a dynamic change of the system behavior may include a modification of the execution speed of its activities and that is driven at the application level.

The paper proposes the model underlying the architectural abstractions by means of UML diagrams enriched with OCL [25] constraints when required. The UML language has been chosen for two main reasons. First, UML is widely used to design concrete systems. Thus, the use of this language already at the analysis phase simplifies the subsequent implementation of a framework reifying the model [26]. Secondly, many timing issues cannot be described by means of languages like OWL (in particular TimeOWL), since they must not convey information to the applications, being related to dynamics only.

The proposed model is applied to a simplified case study dealing with the process of beer making. The case study, even though simple from an applicative point of view, turns out to be significant with respect to the timing requirements that must be met.

The paper is organized as follows: Section 2 introduces the temporal model; Section 3 presents a case study dealing with the artisanal production of beer and how the proposed temporal model may help in realizing some of the major aspects of the problem; finally, Section 4 presents conclusions and outlines future work.

2 The Temporal Model

This section presents the temporal model formalizing its constituting elements, their properties and their relations by means of UML diagrams (both static and dynamic) enriched with OCL constraints when needed.

2.1 Basic Performer Types

*Time Sensitive Performer*s are entities that *perform* domain-related activities and that are someway related to the concept of time. As sketched in Figure 1, *Time Sensitive Performer*s are classified according to the relation they have with the concept of time, and precisely:

- A *Time Conscious Performer* needs to reason on facts placed in a temporal context (*Timed Fact* in Figure 1), without any reference to when such a reasoning is actually realized. The typical example of a *time conscious* activity is the off-line analysis of time-stamped data
- A *Time Observer Performer* needs to read current time from a clock (*Clock* in Figure 1). Again, this property is in no way related to the mechanism that triggers the execution of the activity. As an example, any activity that samples and stores data with a timestamp is a *time observer* activity
- Finally, the only specialization that concerns activation mechanisms is the *Time Driven Performer*, reifying a generic activity whose execution is triggered at specified time instants. This definition includes both periodic and

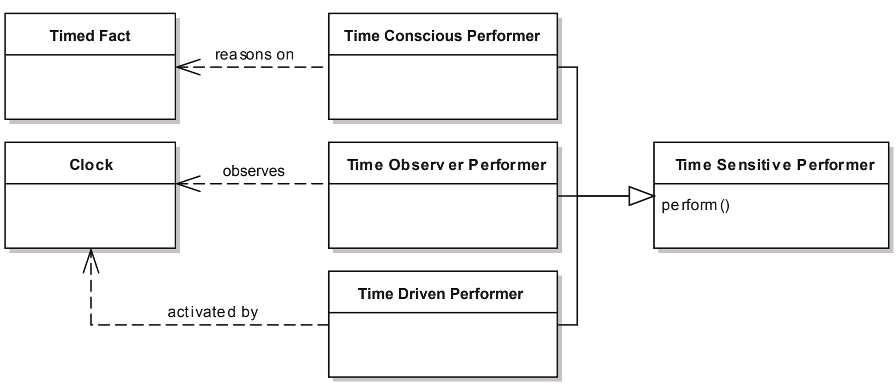


Fig. 1. Performers and their relations with time concepts

aperiodic activations. An example of *time driven* activity is the periodic sampling of data by a sensor.

The following subsections present the time-related abstractions needed to fulfill the requirements all the basic performer types claim.

2.2 Time Consciousness

In any actual software system, time is necessarily discrete. The indivisible time quantum is named *Grain*. A *Timeline* is a sequence of grains, labeled by consecutive integer numbers. The number associated with each grain is stored in the *value* attribute. A *Time Interval*, defined on a timeline, is a subset of consecutive grains belonging to that timeline.

Being *Fact* a domain-dependant data, a *Timed Fact* is a fact associated to a time interval representing the fact's temporal context. All the introduced concepts are sketched in Figure 2.

The concepts related to time consciousness would be suitable to be represented by means of TimeOWL. However, using two formalisms to describe the model we propose would generate confusion. Therefore, UML has been chosen as the single modeling language, since, as previously anticipated, it is able to express all the concepts underlying the model.

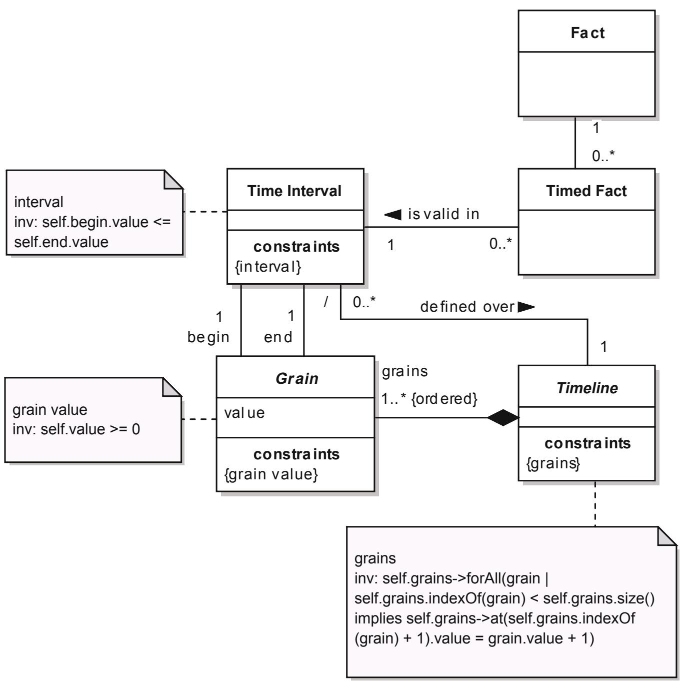


Fig. 2. Concepts related to time consciousness

2.3 Time Observability

Figure 3 sketches the abstractions related to the concept of time observability. In particular, a *Timer* is a source of events that are interpreted as equally spaced in time. Timers can be arranged in a hierarchy, in which every descendant timer has exactly one reference timer. The root of this hierarchy is the *Ground Timer*, which generates events that are understood as marking the flow of the real external time. On the other hand, a *Virtual Timer* is a kind of timer that counts (through its *internal counter*) the number of events it receives from its reference timer and in turn generates events every time the number of received events equals its *period*. This behavior is shown in the state diagram presented in Figure 4. A particular case of *Periodic Entity* (i.e., entities interested in receiving events generated by a timer) is the *Clock*, whose purpose is to keep track of current time on a timeline. Every time a clock receives an event generated by its timer, it advances the current time (*current time*) on its timeline, as depicted in Figure 5.

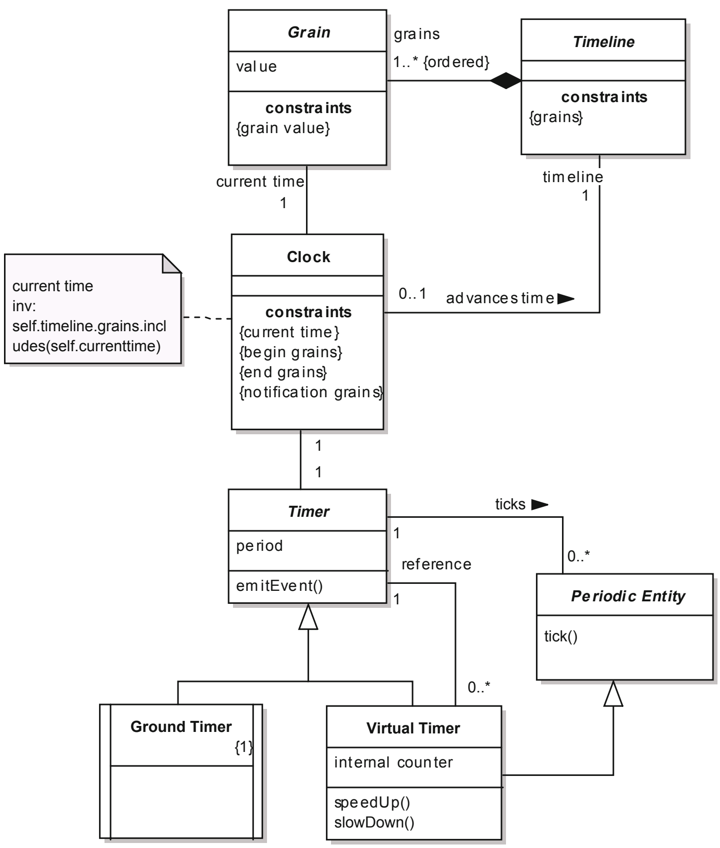


Fig. 3. Concepts related to time observability

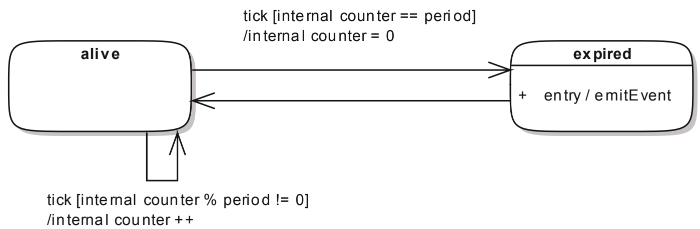


Fig. 4. Virtual timer behavior

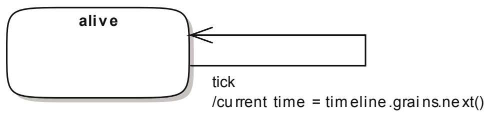


Fig. 5. Clock behavior

2.4 Time Triggering

A *Clock* is an entity that on one hand can be used by time observer entities to read current time, and on the other hand can be used to trigger the execution of *Time Driven Performers*. It is useful to remark again that *Time Conscious Performers* and *Time Observer Performers* are activated by some generic entities unrelated to time. The additional concepts that underlie the activation of *Time Driven Performer*s by *Clock*s are shown in Figure 6. A clock is possibly associated to a set of *Time Driven Performer*s that must all be always activated together at predefined time grains, specified by the clock's *notification grains* attribute. In order to allow the use of the defined concepts for the definition of standard hard real-time systems, it is useful to associate to every *Time Driven Performer* a list of *Performer Execution* entities, each composed of a *begin* time grain (the instant when the performer's execution can be triggered) and an *end* grain (representing the deadline for that particular execution of the performer). Of course, several constraints are required to guarantee the consistency of the picture:

- For every execution of a performer, the deadline must obviously be in the future with respect to the corresponding activation. That is, the value of the *end* attribute of every *Performer Execution* must be greater than or equal to the value of the corresponding *begin* grain
- The set of *begin* grains for all performers must be the same (since the clock activates all the associated performers at every emitted event), and must coincide with the set of the clock's *notification grains*

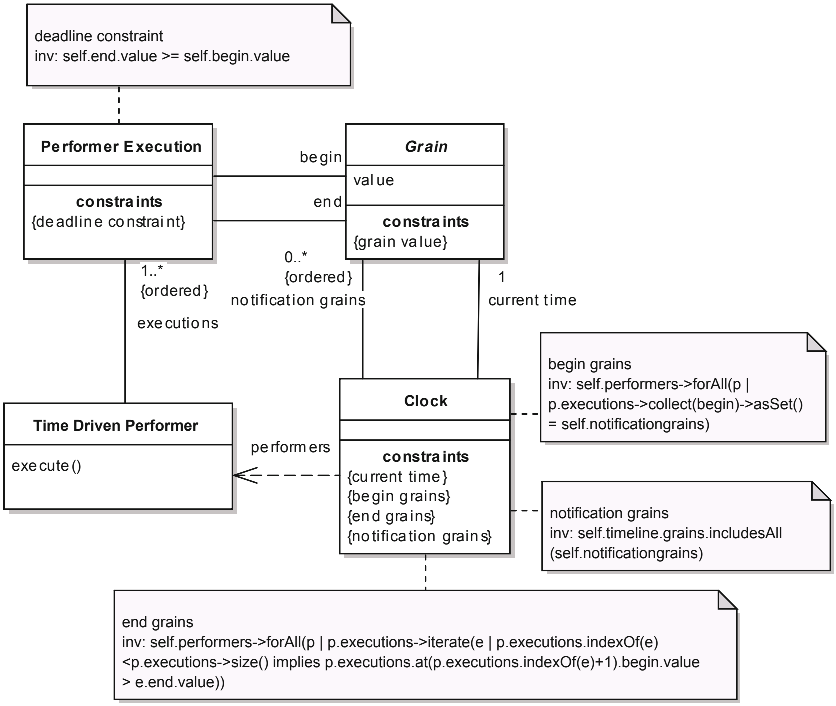


Fig. 6. Concepts related to time triggering

– To ensure that every execution of a performer is terminated before the next one is triggered, for every performer and for every execution of that performer, the value of the *end* grain must be less that the value of the *begin* grain of the next execution, if present

Note that the particular case of periodic activities is included in this description. In fact, a periodic activity can be reified by a *Time Driven Performer* with the following properties:

– The set of *executions* is unbounded
– Every execution is characterized by the same difference between the values of the *end* and *begin* grains
– The *begin* grains are equally spaced in time. Of course, the *notification grains* of the associated clock must share this property too

The details of the management of performer executions by a clock are shown in the sequence diagram of Figure 7. Every time a clock is ticked by its timer, it first advances the time it is keeping. Then, it checks if the current time is contained in the set of *notification grains*, that is, if it must trigger its associated performers at this time. If this is the case, the clock triggers the execution of every associated performer by invoking the corresponding *execute* action.

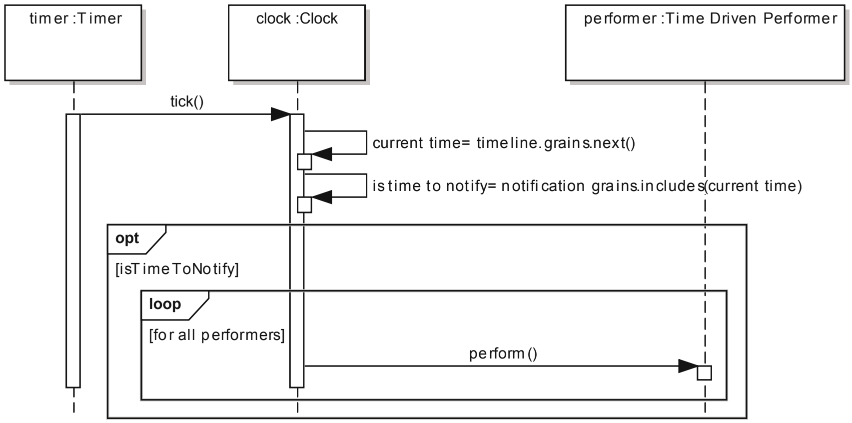


Fig. 7. Clock behavior

The state diagram of a *Time Driven Performer* is presented in Figure 8. The performer is left in its *waiting* state until the associated clock triggers an execution. The performer then goes to the *running* state, whose entry point is the performer's *perform* action. Once the performer completes its execution, it goes back to the *waiting* state until the next execution is triggered.

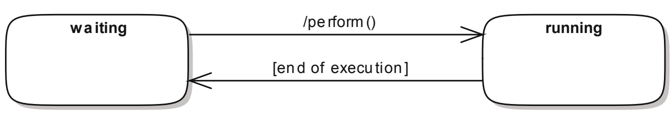


Fig. 8. State diagram of a time driven performer

An interesting feature of the model is the capability to dynamically vary the activation speed of time driven performers at run-time. This requires that the clock advance speed can be modified. At the aim, the two actions *speedUp* and *slowDown* have been defined for a *Virtual Timer* that respectively decrease and increase the timer's period.

2.5 Advanced Performer Types

More complicated performer behaviors can be realized by combining two or more of the time-related properties of Figure 1, as sketched in Figure 9. Some care must be used to guarantee consistency when designing entities that are both time driven and time conscious. In fact, it is desirable that the behavior of all the performers that are triggered at the same grain (either because they are connected to the same clock or because the corresponding clocks activate them at the same time) does not depend on the order in which the executions are

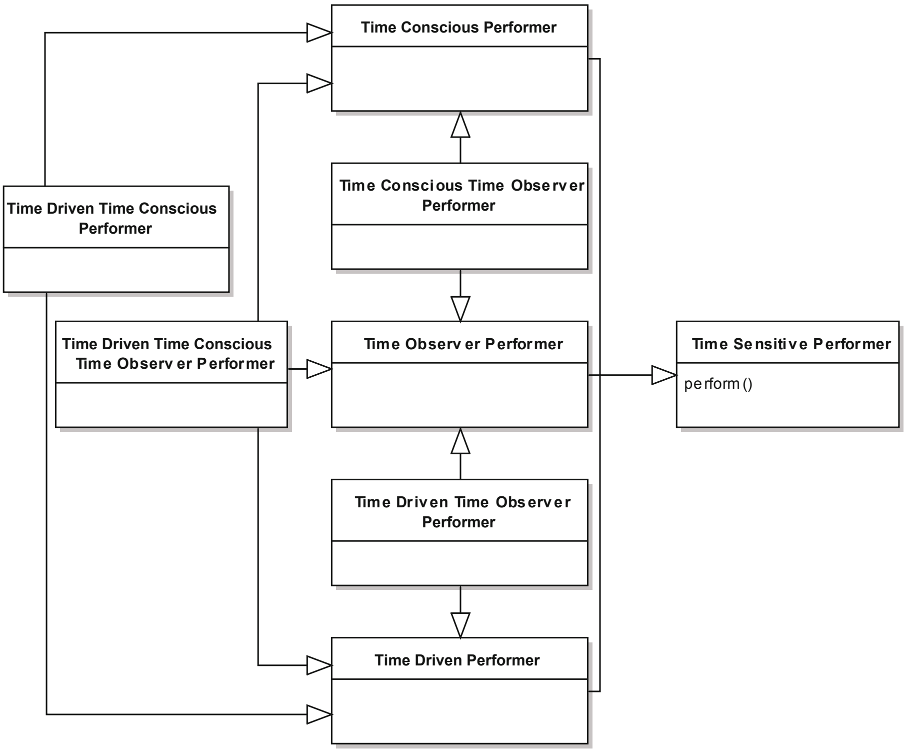


Fig. 9. Advanced performer types

actually managed, which depends on low-level details such as the number of available cores or the particular scheduling algorithm that is being used. Therefore, it is necessary to guarantee that all the time conscious performers that are triggered simultaneously share the same view of the timelines in which they are interested, to avoid the situation of a performer that reads timed facts written by a performer triggered at the same time just because the latter was granted higher execution priority by the low-level scheduler.

Since every *Performer Execution* is characterized by a time interval delimited by the *begin* and *end* grains, a possible solution for this consistency problem is that all performers read timed facts at the beginning of the execution interval and write timed facts only at the end of the interval (i.e., at the deadline for that execution), even if the actual execution ends before the deadline. The performer structure that realizes this mechanism is described by the state diagram of Figure 10. A time driven time conscious performer is at first in the *waiting* state. When the associated clock triggers an execution (by invoking the *execute* action), the performer goes to the *reading* state, where it can read timed facts from timelines by means of its *observe* action. Once this phase has been completed, the performer spontaneously goes to the *running* state, analogous to the homonymous state for a time driven performer (Figure 8). At the end of

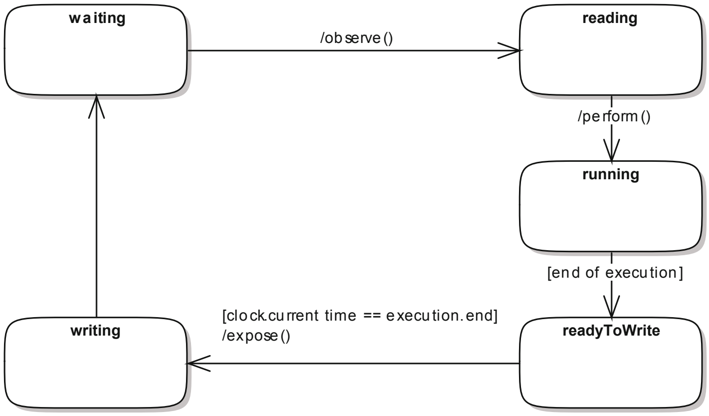


Fig. 10. State diagram of a time conscious time driven performer

the execution, instead of directly writing timed facts to timelines, the performer goes to the *readyToWrite* state, where it stays until the end of the interval for the current execution. Once the deadline is reached, the performer goes to the *writing* state, where it writes timed facts by means of the *expose* action. Once the writing operation terminates, the performer goes back to the *waiting* state until the next execution is triggered.

In an actual implementation, the concrete component in charge of the management of performer executions must thus guarantee that when the execution of a set of performer is triggered, all the performers of the set read timed facts before any is allowed to start the actual execution, and that every performer writes timed facts only at the end of the validity interval for its execution.

3 A Case Study: The Brewing Process

This section introduces a simplified example of how the concepts introduced in Section 2 can be easily applied to build a time-aware system. The case study concerns an automated plant for artisanal production of beer. Even though it may appear a simple application, it is particulary significative from the point of view of the involved temporal constraints and requirements.

3.1 Description

The process of making beer is known as *brewing*. Its purpose is to convert starch source (barley malt) into a sugary liquid called wort, that will be boiled with hops and afterwards transformed into an alchoolic beverage by yeast during fermentation. Figure 11 shows a typical brewing configuration, which includes three tuns, used for different purposes during the brewing. The process includes the following three macro-phases:

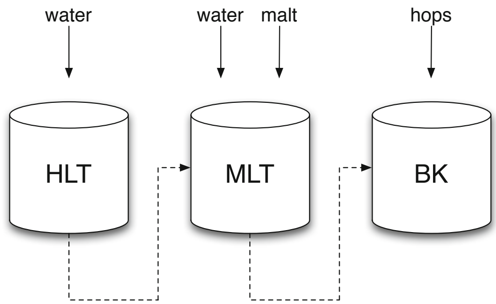


Fig. 11. Typical brewing configuration

- *Mashing* is the process in which milled malted barley is mixed with hot water inside MLT (Mash Lauter Tun) at different temperatures for different time intervals, during which the temperature must be kept constant (e.g., 52 ℃ for 10 minutes, 66 ℃ for 60 minutes, 78 ℃ for 15 minutes)
- *Lautering* aims at transferring the wort obtained after mashing into BK (Boil Kettle), separating solid grains from liquid wort (this is achieved using a false bottom placed in MLT). During this phase, the grains can be washed with hot water coming from HLT (Hot Liquor Tank), to extract more sugar from them (*Sparging*)
- *Boiling* is the last phase during which the wort collected in BK must be boiled (usually for 90 minutes). Besides, one or more hops are added as a source of bitterness, flavor and aroma at different time intervals (e.g., hop 1 after 10 minutes of boiling, hop 2 after 60 minutes, hop 3 after 90 minutes).

3.2 Modeling

In the process described above it is possible to identify at least one *time-related* aspect for each phase of the brewing process that can be addressed using the proposed temporal model:

(I) Boiling: hops need to be added at fixed times during boiling. This is an example of an aperiodic real-time activity. As shown in Figure 12, a *timer* T1, having the ground timer GT as its reference, is given acting as the source of ticks for a *clock* C1, which drives a *time driven performer* P1 that throws hops.

(II) Lautering: while transferring the liquid wort from MLT to BK (and from HLT to MLT if sparging is required), fluid volume sampling must be speeded up in order to provide better feedback. This is an example of a periodic real-time activity whose activation speed needs to be tuned. As shown in Figure 13, a *timer* T2 is given acting as the source of ticks for a *clock* C2, which drives a *time driven performer* P2 that acquires volume samples with different periods (e.g., 500 ms during the lautering (and sparging) phase, 30 s otherwise).

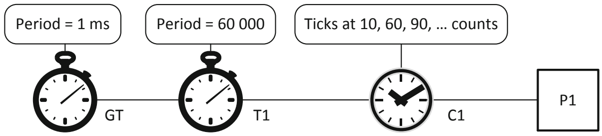


Fig. 12. Example of an aperiodic activity

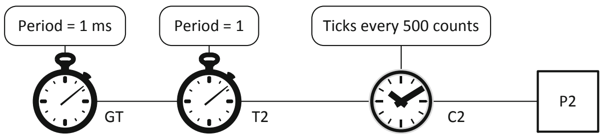


Fig. 13. Example of a periodic activity

(III) Mashing: during mashing, the MLT temperature must be periodically sampled and filtered in order to control the heating system to keep temperature constant. These are respectively examples of time conscious time observer and time conscious activities. As shown in Figure 14, a *timer* T3 is given, acting as the source of ticks for a *clock* C3, which drives a *time driven time conscious time observer performer* P3 that acquires MLT temperature samples with a fixed period (e.g., 500 ms), and saves them in timed facts on a *timeline* TL with the timestamp read from C3. Also, T3 acts as the source of ticks for a second *timer* T4, which in turn acts as the source of ticks for a *clock* C4, which drives a *time driven time conscious performer* P4 that reads from TL the not yet processed samples and computes the corresponding mean with a fixed period (e.g., 15 s).

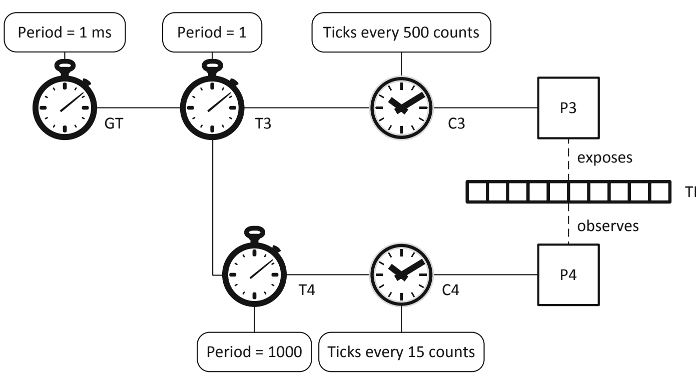


Fig. 14. Example of the use of timelines

4 Conclusions and Ongoing Activities

The paper presents a model that captures both the static and the dynamics aspects of time-related concepts. The model has been applied to design a plant for the artisanal production of beer because of its temporal requirements that well fit with the proposed model.

The temporal abstractions have been reified in an architecture implementation framework [27] developed in the Java programming language. Even though the language requires a virtual machine, the implementation allows the construction of soft real time systems thanks to the explicit management of the actual running activities performed by the implemented scheduler.

The framework has been successfully exploited for the achievement of the experimental results [28] related to a publish/subscribe platform denoted Space Integration Services [29]. In particular, a set of time driven performers has been implemented that periodically produces a set of publications. A time conscious time observer performer captures such publications and places them in a timeline. Another time conscious time observer performer captures the time in which notifications are ready by placing them in another timeline. Finally, an off-line time conscious performer is in charge of correlating the timed facts of the two timelines.

The main future development of the ideas presented in this paper concerns the re-implementation of the framework in a language that is more suitable for testing in hard real-time applications, such as C/C++. This new implementation could be used for the development of hard real-time applications either in the form of bare-metal, os-less programs for embedded systems, or to be executed on real-time operating systems.

References

1. Shaw, M., DeLine, R., Klein, D.V., Ross, T.L., Young, D.M., Zelesnik, G.: Abstractions for software architecture and tools to support them. IEEE Transactions on Software Engineering 21(4), 314–335 (1995)
2. Kristensen, B.: Architectural abstractions and language mechanisms. In: Proceedings of the Software Engineering Conference, Asia-Pacific, pp. 288–299 (1996)
3. Garlan, D., Perry, D.: Introduction to the special issue on software architecture. IEEE Transactions on Software Engineering 21(4), 1–6 (1995)
4. Shaw, M., Garlan, D.: Software Architecture: Perspectives on an Emerging Discipline. Prentice Hall (1996)
5. Fiamberti, F., Micucci, D., Tisato, F.: An architecture for time-aware systems. In: 2011 IEEE 16th Conference on Emerging Technologies & Factory Automation (ETFA), pp. 1–4. IEEE (2011)
6. W3C: OWL 2 Web Ontology Language, http://www.w3.org/TR/owl2-overview/
7. W3C: RDF Primer, http://www.w3.org/TR/rdf-primer/
8. O'Connor, M.J., Das, A.K.: A Method for Representing and Querying Temporal Information in OWL. In: Fred, A., Filipe, J., Gamboa, H. (eds.) BIOSTEC 2010. CCIS, vol. 127, pp. 97–110. Springer, Heidelberg (2011)

9. Gutierrez, C., Hurtado, C.A., Vaisman, A.: Introducing time into RDF. IEEE Transactions on Knowledge and Data Engineering 19(2), 207–218 (2007)
10. Tappolet, J., Bernstein, A.: Applied Temporal RDF: Efficient Temporal Querying of RDF Data with SPARQL. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 308–322. Springer, Heidelberg (2009)
11. Krieger, H.-U.: Where Temporal Description Logics Fail: Representing Temporally-Changing Relationships. In: Dengel, A.R., Berns, K., Breuel, T.M., Bomarius, F., Roth-Berghofer, T.R. (eds.) KI 2008. LNCS (LNAI), vol. 5243, pp. 249–257. Springer, Heidelberg (2008)
12. Hobbs, J.R., Pan, F.: An ontology of time for the semantic web. ACM Transactions on Asian Language Information Processing (TALIP) - Special Issue on Temporal Information Processing 3, 66–85 (2004)
13. W3C: Time Ontology in OWL, http://www.w3.org/TR/owl-time/
14. Lutz, C., Wolter, F., Zakharyashev, M.: Temporal description logics: A survey. In: 15th International Symposium on Temporal Representation and Reasoning (TIME 2008), pp. 3–14. IEEE (2008)
15. Buckl, C., Gaponova, I., Geisinger, M., Knoll, A., Lee, E.A.: Model-based specification of timing requirements. In: Proceedings of the Tenth ACM International Conference on Embedded Software, EMSOFT 2010, pp. 239–248. ACM, New York (2010)
16. Zhao, Y., Liu, J., Lee, E.A.: A programming model for Time-Synchronized distributed Real-Time systems. In: 13th IEEE Real Time and Embedded Technology and Applications Symposium (RTAS 2007), pp. 259–268. IEEE (2007)
17. OMG: MARTE Modeling and Analysis of Real-Time and Embedded systems
18. OMG: UML Profile for Schedulability, Performance, and Time
19. Group, O.M.: OMG Unified Modeling LanguageTM (OMG UML), Superstructure, http://www.omg.org/spec/UML/2.4.1/
20. SAE: AADL Architecture Analysis and Design Language (2009)
21. de Niz, D.: Diagrams and Languages for Model-Based Software Engineering of Embedded Systems: UML and AADL
22. Henzinger, T., Horowitz, B., Kirsch, C.: Giotto: a time-triggered language for embedded programming. Proceedings of the IEEE 91(1), 84–99 (2003)
23. Gamatié, A., Gautier, T., Guernic, P.L., Talpin, J.P.: Polychronous design of embedded real-time applications. ACM Trans. Softw. Eng. Methodol. 16(2) (2007)
24. Lopez, P., Medina, J., Drake, J.: Real-time modelling of distributed component-based applications. In: 32nd EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA 2006), pp. 92–99 (2006)
25. Group, O.M.: Object Constraint Language, http://www.omg.org/spec/OCL/2.3.1
26. Fiamberti, F., Micucci, D., Tisato, F.: An Object-Oriented Application Framework for the Development of Real-Time Systems. In: Furia, C.A., Nanz, S. (eds.) TOOLS 2012. LNCS, vol. 7304, pp. 75–90. Springer, Heidelberg (2012)
27. Taylor, R.N., Medvidovic, N., Dashofy, E.M.: Software Architecture: Foundations, Theory and Practice, 1st edn. Wiley (2009)
28. Bernini, D., Fiamberti, F., Micucci, D., Tisato, F.: Architectural Abstractions for Spaces-Based Communication in Smart Environments. Journal of Ambient Intelligence and Smart Environments, Thematic Issue A Software Engineering Perspective on Smart Applications for AmI (in Press)
29. Bernini, D., Micucci, D., Tisato, F.: A platform for interoperability via multiple spatial views in open smart spaces. In: The IEEE Symposium on Computers and Communications, Riccione, Italy, pp. 1047–1052 (2010)

Toward a Perdurantist Ontology of Contracts

Sergio de Cesare[1] and Guido L. Geerts[2]

[1] Department of Information Systems and Computing,
Brunel University,
Uxbridge, UB8 3PH, U.K.
sergio.decesare@brunel.ac.uk
[2] Department of Accounting and MIS,
Alfred Lerner College of Business and Economics,
University of Delaware,
Newark, DE 19716, U.S.A.
geerts@udel.edu

Abstract. Contracts are fundamental toward characterising the very nature of a firm (or enterprise). The firm is considered by some economic theories as a bundle of contracts and contracts in turn are considered also as bundles of rights and obligations (commitments). As such it can be argued that the ontological relationships between the firm and its contracts can be explained through a set of mereological (or whole-part) relationships. Specifically, the relationships between a contract and its parties and between the parties and their rights/commitments are all mereological. This view of what contracts are may appear at first surprising but a perdurantist interpretation of contracts results in such an ontology. The main contribution of this paper is a perdurantist ontology of contracts which introduces the following distinctive features: (1) a differentiation between contract specification and contract execution, (2) contract executions as objects whose spatio-temporal extents intersect those of its parties and (3) a generic model of contractual commitments and fulfilment events impacting the economics of the enterprise. The ontology proposed in this paper is applied to an example scenario to demonstrate its benefits in enterprise modelling.

Keywords: Enterprise Modelling, Contracts, Ontology, Perdurantism.

1 Introduction

Contracts regulate the life of a firm. The agreements that organisations stipulate with other parties (consumers, suppliers, etc.) determine the way in which business processes are executed, the products and services offered, and the economic success or failure of the enterprise. All organisations become party to contracts continuously. Agreeing to a contract can be verbal and implicit (e.g., a baker selling a loaf of bread) or written and explicit (e.g., the supply of oil over 20 years). However implicit or explicit, short- or long-lived the agreed commitments are, contracts are fundamental toward defining the relationships of the enterprise with other economic agents.

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 85–96, 2012.

The defining role of contracts is emphasised by several economic theories of the firm albeit with a focus on different specific aspects. Transaction Cost Theory emphasises the costs derived from negotiating and renegotiating contracts from which a firm's transactions with external as well as internal parties (e.g., employees) arise [1]. Agency theory views the firm as a nexus of contracts between resource holders. The focus here is on service contracts in which agents are hired and on the costs associated with such agency agreements. More explicitly Reve [2] and Ayotte and Hansmann [3] view the firm as a bundle of (internal and external) contracts. The reasons for adopting such a view of the firm is especially made clear by Ayotte and Hansmann [3] who underline that the firm cannot be merely considered as a set of owned assets since, especially in recent times, the value of certain companies, like Netflix, is based on their "assemblage of contractual relationships" (p. 5). In fact Netflix does not own the products (e.g., DVDs) they rent.

Similarly contracts are central to Accounting theories such as Commitment Accounting whereby the rights and obligations arising from enforceable agreements should be recorded as soon as the agreement is reached [4]. Ijiri's work [4], among others, strongly influenced the Resources-Events-Agents framework (REA). REA is a conceptual modelling framework specialised in the representation of economic events underlying the life of a business organisation, the interactions that arise among agents taking part in such events and the resources that are exchanged, acquired, converted or produced as a result of such events. REA was originally conceived by McCarthy [5] as a generic conceptual model aimed at improving the way traditional accounting methods represented an organization's transactions. Subsequently REA was reinterpreted to provide it an ontological grounding [6] and extended to introduce new concepts [7] [8] and refine existing ones. In relation to the research presented in this paper the most significant extensions relate to the concepts of commitment and contract [8]. Commitment is now considered a fundamental REA construct while contract is an Enterprise Modelling pattern that can be applied to generate business applications. REA is now a well-recognised Enterprise Modelling approach.

In the field of Enterprise Modelling REA is not unique in leveraging the notion of agreed commitments to model the business and its processes. In REA economic events represent fulfilments of contractual commitments. While the concept of contract (as a whole) was always implicitly present in REA and then defined with Hruby [8], a further modelling approach called Approach Based on Contract (ABC) [9] explicitly adopts contracts as the starting point to model business processes. Overall ABC, similarly to REA, views contracts as a means to specify service-provider relationships involving, for example, an exchange, the possible production of goods and/or the provision of services.

This paper presents research aimed at developing an Ontology of Contracts as an example of the challenges encountered when modelling socially constructed objects. While our ontology utilises theories and approaches, as the ones above, as sources to inform, drive and clarify the notion of a contract, the model that we propose, unlike previous work, adopts a philosophical theory of existence known as perdurantism (or 4D).

The paper is organised as follows. Section 2 defines ontology and presents the reasons for adopting perdurantism. Section 3 provides an overview of the foundational 4D ontology adopted to drive the discovery of the contract domain ontology. Section 4 presents the proposed perdurantist Contract Ontology and Sections 5 and 6 will discuss the limitations of the ontology proposed and future work.

2 Ontology

The research in this paper is inspired by philosophical ontology. In Philosophy, ontology is defined by Lowe as “the set of things whose existence is acknowledged by a particular theory or system of thought” [10]. While this definition is not widely adopted among the Information Systems (IS) and Computer Science communities, it does remind ontology engineers of an important principle. This principle is that of ontic (or ontological) commitment. Lowe’s definition indicates that any ontological model must refer to things that exist in reality and, as a consequence, that there must be evidence of such things existing, evidence that could derive, for example, from an analysis of information systems data or scientific data.

Conversely, Gruber [11] defines ontology as a (formal) “specification of a conceptualisation”. This widely referenced definition also has its merits. First, it emphasises the fact that any model created for the purposes of information systems development or knowledge representation must be expressed in a formal language in order for it to be processed by computers. Second, it implicitly brings into focus how conceptualisations naturally develop in the minds of software and knowledge engineers as part of the initial phases of IS development. The major differences with Lowe’s definition also lie in the two points just made.

In fact a formal specification relies on the logical consistency of formal semantics thus assuring that the statements made in the model are not contradictory. Logical consistency is essential for executable models but at the same time it is also necessary that an ontological model be a model of reality. Therefore alongside formal semantics, an ontological model must possess a high degree of real world semantics achieved by accurately mapping things in the real world to things (symbols) in the model. Logical consistency without an accurate representation of reality is likely to produce ineffective and costly to maintain enterprise systems. Consequently if one only formally models a ‘conceptualisation’ rather than reality, then there is a risk that the models produced become representations not of the real world but of “concepts conceived as human creations” as Smith [12] aptly states.

The development of accurate ontological models is difficult and an enterprise modeller would instinctively begin from the representation of his or her own conceptualisations of the organisational domain. Particularly challenging is the modelling of socially (or intentionally) constructed objects, i.e. things that exist because human society has created them [13]. Examples include governmental institutions, money, marriage and a long list of social constructs including, of course, contracts. Enterprise modelling relies heavily on socially constructed objects. These objects are indeed human creations and therefore manifestations of human conceptualisations. One may argue that ontologies of domains like law, economics and the firm mostly model conceptualisations since there does not appear to be any real tangible counterpart in the world (e.g., one cannot touch the agreement underlying a contract). Instead, as this paper will demonstrate in the case of contracts, socially constructed objects are real and produce real world objects (mainly in the from of states and events) that can be accurately modelled. Philosophical ontology can help to produce accurate enterprise models.

In Philosophy, ontology, as a discipline, is the study of existence and of the kinds of things that exist. There are two predominant theories: endurantism and perdurantism [14]. The main difference lies in each theory's conception of time and how objects change in time. 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). 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). In perdurantism an object has a four-dimensional extension (or extent) in the universe (i.e., the region of space-time that it occupies) and it is not therefore totally present at any given instant, but instead only partially present. Identity is defined by 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 dissimilar 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 is partially present at any time or portion of its spatiotemporal extension.

The analysis and interpretation of contracts carried out in this paper is based on perdurantism. There are some fundamental reasons for adopting such a theory over endurantism especially for the purposes of modelling the enterprise. As stated above, enterprise ontologies require the representation of many socially constructed objects. These objects live and evolve with the enterprise through a series of complex events that define the interactions occurring among parts of the organisation and between the organisation and external parties. These events often produce objects and/or states, some of which overlapping, and together (events and states) define the processes that occur during an organisation's existence (with contracts playing a significant role in defining such processes). All this requires an ontological theory that is capable of more naturally representing events, states (more generally temporal parts) and overlapping objects. As explained above, objects in perdurantism extend through time and are therefore intrinsically capable of having spatiotemporal extensions that contain or overlap with the extensions of other objects. Such temporal containment or overlap becomes very difficult or impossible to model with an endurantist ontology.

In order to develop a domain model, such as the Contract Ontology of Section 4, it is necessary to adopt a foundational ontology that not only clearly answers the question of what it means for a thing to exist but also defines the kinds of existence that things can have (i.e., a categorical theory). This research adopts the Business Object Reference Ontology (BORO), as its foundational ontology.

3 A Perdurantist Foundational Ontology

BORO, developed by Partridge [15], is a perdurantist upper level ontology strongly based on extensionality. BORO inspired the upper level ontology of the International Defence Enterprise Architecture Specification for exchange Group [16] and 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 Contract Ontology described in the following section. It is beyond the scope of this paper to provide an exhaustive explanation and definitions of the whole foundational ontology. The discussion will be limited to the higher level and scoped to those foundational classes and relationships that will be necessary to model contracts. For an in depth presentation of BORO the reader is invited to refer to Partridge [15] in its original form or IDEAS [16] for a slightly modified, yet still detailed, version.

Figure 1 presents a graphical representation of the foundational ontology. The notation is that of the Unified Modelling Language (UML). For limitations of space and in order to provide the reader with an uncluttered picture of the ontology, the diagram only presents a partial view of the entire foundational ontology; a more complete representation is provided by IDEAS [16].

At its highest level the BORO foundational ontology represents:

- *Objects*: Anything that exists. (In IDEAS the term *Thing* is used in place of *Objects*.)
- *Individuals*: An individual 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 (*IndivudalTypes*), type instances (*Powertypes*) or tuple instances (*TupleTypes*). Only *TupleTypes* are explicitly represented in Figure 1.
- *Tuples*: A tuple is a relationship between two or more objects.
- *Tuple Types*: A type whose instances are 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 individuals in general) are bounded by events. A state can have further temporal parts (i.e., states and events).
- *causedBy*: This tuple type represents the relationship between an event and the individual(s) which causes the event.
- *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 individual(s) whose creation is triggered by the event.
  - *dissolves*: Relates a dissolution event with the individual(s) whose dissolution is triggered by the event.
- *happensAt*: This tuple type relates an event with a *TimeInstant* 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 person named *John* who enrols on an undergraduate programme becoming a student (Figure 2). As the figure shows *John* (as a 4D individual) extends through space-time. A portion of *John*'s extension has a temporal part named '*John's UG Student State*' which is created by an event named '*John's Enrolment*' and finishes (or is dissolved) by another event named '*John's Graduation*'. The two events and the state illustrated in the figure are all temporal parts of John. Although not represented in the figure, '*John's UG Student State*' can be further decomposed into substates with their respective boundary events. These states may be '*John's 1st year state*', '*John's 2nd year state*', etc.

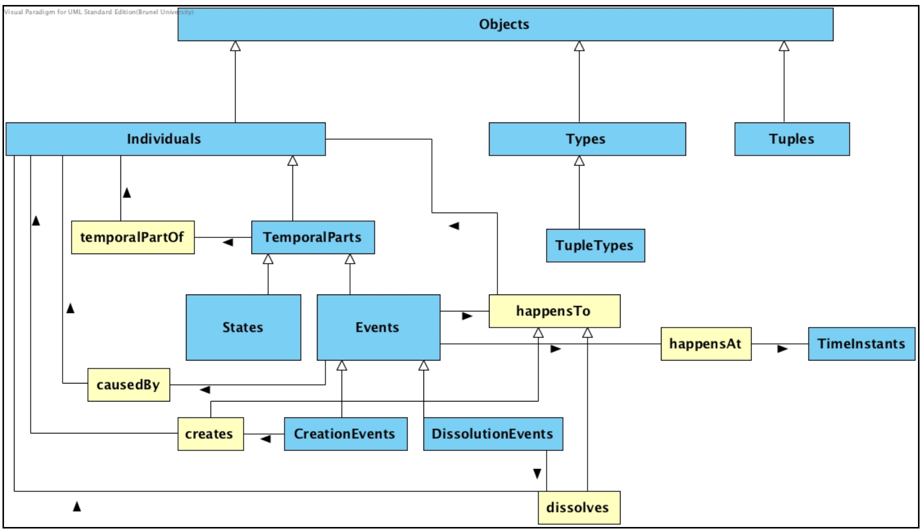


Fig. 1. BORO foundational ontology (partial view). (*Types* are represented in blue or darker colour and *TupleTypes* in yellow or lighter colour).

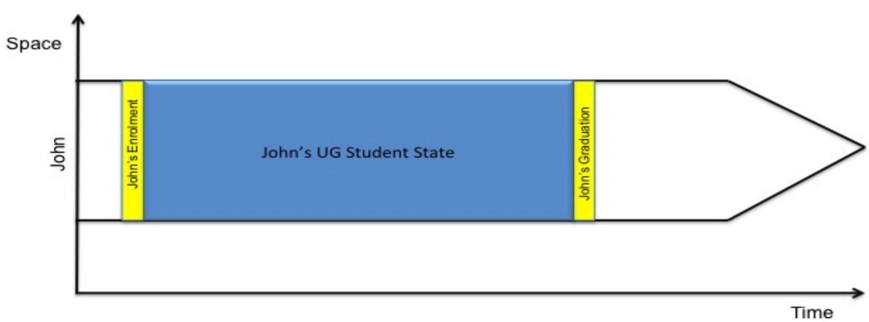


Fig. 2. Example space-time map

4 A Perdurantist Ontology of Contracts

Contracts have been defined and regulated over the centuries by national legislations. While there are some differences in the subtleties of their definitions and the types of contracts regulated, it is possible for the purposes of this research to extract some common elements that all contracts possess. Thereafter, in order to develop the ontology of contracts, it is necessary to semantically interpret these common elements and contracts as a whole from the perspective of the foundational ontology adopted.

Generally speaking a contract can be defined as follows: “an agreement with specific terms between two or more persons or entities in which there is a promise to do something in return for a valuable benefit known as consideration. [...] The existence of a contract requires finding the following factual elements: a) an offer; b) an acceptance of that offer which results in a meeting of the minds; c) a promise to perform; d) a valuable consideration (which can be a promise or payment in some form); e) a time or event when performance must be made (meet commitments); f) terms and conditions for performance, including fulfilling promises ...” (Definition from http://dictionary.law.com/ also cited by Kabilan [17]).

Dissecting the definition it appears that the necessary elements of a contract are:

1. Agreement among persons: At least two persons or entities must consent to the specific terms of the contract. An entity (or juridical person) is accorded ‘legal personality’ and therefore considered by law a person. Hence, we will refer only to persons. When agreeing to a contract a person becomes party to the contract. The agreement is formed once both (or all) parties give their consent (e.g., verbally or by signature). Elements (a) and (b) in the definition together form the agreement. Once the offer is accepted then the contract is stipulated.
2. Promise: The parties commit themselves to fulfilling obligations (or commitments) according to the terms agreed. These commitments can be of different types (generally either to perform or to not perform). Normally there is a relationship of reciprocity between the set of commitments to which both parties agree. A similar type of relationship (duality) exists between the fulfilments of the commitments.
3. Consideration: In general an exchange of resources (in the economic sense) that is of value to the parties.

From the above brief analysis it is possible to begin a perdurantist interpretation of contracts by discovering the spatiotemporal extensions of a contract, its parts and the persons party to it. The question to ask is: what is the spatiotemporal extension of a contract? In attempting to answer this question it becomes apparent that the term contract bundles two different and related meanings. First, contract refers to the actual written or verbal specification that documents the agreement and specifies all the terms and conditions (in general *ContractClauses*) that the parties must respect. Second, contract refers to the execution of the events that occur after the stipulation in fulfilment of the contractual obligations. As such two distinct types are identifiable:

ContractSpecifications and *ContractExecutions*. The former is manifest in the case of written contracts whereby a document models, among other things, the commitments and the future fulfilment events. A contract execution realises a contract specification. The extension of a contract specification is straightforward in the case of a written document and it represents the life of the document itself starting from its stipulation.

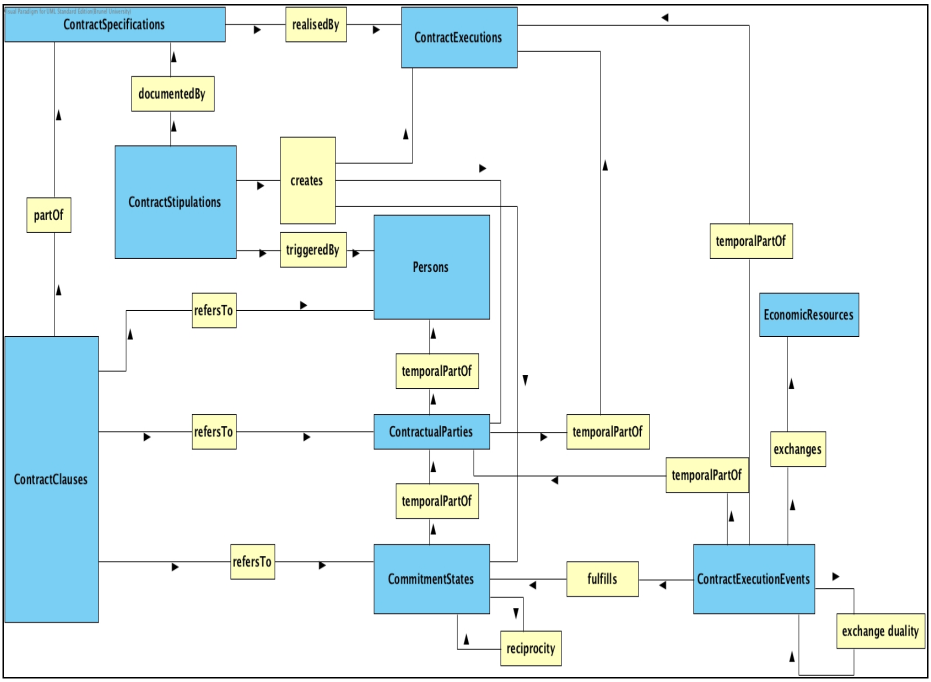


Fig. 3. Partial Perdurantist Ontology of Contracts

ContractStipulations are significant creation events. It is with the contract stipulation that the execution begins and the committed *Persons* enter in a state of being *ContractualParties*. At the same time the agreed obligations place the parties in various *CommitmentStates* each of which will terminate (or dissolve) once each individual commitment is fulfilled. Commitments are fulfilled via *ContractExecutionEvents*. It is at this point, once we start analysing the type of relationships between the instances of the types defined above, that the perdurantist model of contracts shows its fundamental differences with other contract representations in the literature. Figure 3 depicts the Perdurantist Contract Ontology.

Figure 4 illustrates such relationships with a space-time map by representing a simple contract in which a Car Dealer (named DMS) agrees to sell Mary a car and Mary agrees to buy the car by paying a certain amount of money in two instalments. In relation to the contract execution the following significant events occur:

(E_1) Stipulation of the contract to sell/buy the car at time t_1. This event creates DMS's and Mary's respective states (P_1 and P_2) of being party to the contract. It also produces three commitment states: DMS's commitment to deliver the car (C_1) and Mary's two commitments to pay the instalments (C_2 and C_3). It can be noted from Figure 2 that P_1 is temporally part of DMS and C_1 is temporally part of P_1. Similarly C_2 and C_3 are temporal parts of P_2 which is a temporal part of Mary.

(E_2) DMS's delivery of the car at t_2 ending DMS's commitment C_1.

(E_3) Mary's first payment at t_3 ending Mary's commitment C_2.

(E_4) Mary's second payment at t_4 ending Mary's commitment C_3. E_4 also dissolves or terminates the contract execution as the final remaining commitment of the contract is fulfilled.

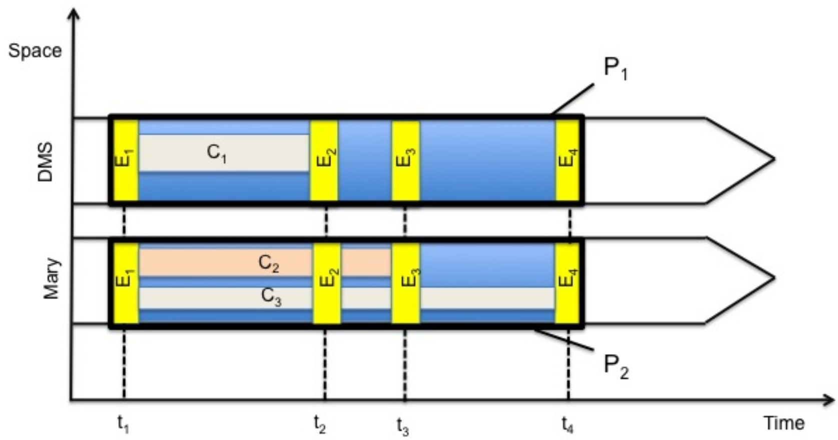


Fig. 4. Space-time map of the execution of a sales contract.

Figure 4 visually shows these temporal part relationships. For example, DMS physically contains its contractual party state, its commitment state and all related events. At this point the answer to the initial question of what is a contract execution appears clear. The execution of the contract (CEx) is the mereological sum of all *ContractualParties* involved. In the example, CEx = $P_1 + P_2$.

5 Discussion

The example in Section 4 serves the purpose of explaining and visually showing how a perdurantist ontology models a socially constructed individual such as a contract or, to be precise, the execution of a contract. Based on this representation being party to a contract is a temporal part of the person agreeing to that contract. In terms of the foundational ontology depicted in Figure 1, *ContractualParties* is a subclass of *States*. An individual contractual party would then also have temporal parts represented by specific *CommitmentStates* and specific *ContractExecutionEvents* fulfilling (or terminating) the commitments. The two contractual parties are together temporal parts of the overall contract execution. Hence, the spatiotemporal extent of a contract execution intersects with the spatiotemporal extents of the persons involved and overlapping with these persons' contractual party states.

The events described above have been simplified in order to keep the example clear. In fact event E_1 may more realistically be decomposed into two further events: the first representing DMS's decision to agree to the contract stipulation and the second representing Mary's decision to agree herself. These two decisions can happen at different times and together would compose the 'complex' stipulation event E_1. Similar considerations can be made for the other events. For example, the buyer can make a payment at a time different to when the seller receives the payment.

Figure 4 also implicitly shows the relationship of *reciprocity* between the set of DMS's commitments (C_1), on the one hand, and the set of Mary's commitments (C_2 and C_3) on the other. Similarly there also exists a relationship of *exchange duality* between both parties' execution events fulfilling their respective commitments, specifically E_2 carried out by DMS versus E_3 and E_4 carried out by Mary. Moreover, the *EconomicResources* exchanged, and not represented in Figure 4, are the car sold and delivered to Mary against the cash that DMS receives as payments. These elements (duality, reciprocity and economic resources) are borrowed from REA and here remodelled.

The ontology in Figure 3 is a partial representation of contracts. A more complete Contract Ontology would require a model that is capable of representing and explaining the set of different alternative scenarios that can unfold once a contract is stipulated, ranging from the case in which all parties comply with the commitments promised to an eventual breach of contract. There are both technical and theoretical considerations to be made. From a technical perspective, it must be considered that some possible scenarios or state-of-affairs (for example, a breach of contract) may not necessarily be described by the contract specification alone, but also by laws and norms that legally integrate contracts of a specific type (e.g., sales contracts) in a mandatory manner. From a theoretical perspective, perdurantism and extensionality must be integrated by a theory capable of explaining alternate possible scenarios. In Philosophy a theory that explains the notion of possibility is that of Possible Worlds (for example, see Kripke [18] and Lewis [19]).

While it is beyond the scope of this paper to provide a detailed representation of contracts based on Possible Worlds or related theories, it would be useful to mention that an integration of perdurantism and Possible Worlds must elegantly answer the question of how states-of affairs of different possible worlds can be mapped to one another and traced back to the same original contract of our actual world. To be consistent with the principle of strong extensionality of the BORO foundational ontology, it would be however appropriate to develop a solution that somehow is rooted itself in extensionality.

6 Conclusion

The Perdurantist Contract Ontology presented in Section 4 represents research that requires further refinement, expansion and evaluation. It represents an example of how a perdurantist ontology is capable of modelling socially constructed objects. Due to limitations of space we have omitted a thorough comparison with similar work aimed at modelling contracts and services (for example, see [17], [20], [21], [22], [23]). Suffice to note here that the main difference lies in the modelling of commitments and their fulfilment events as temporal parts of the parties involved in the contract. We realise that such a representation may radically depart from the common conception of contract whether it be by the technical information systems community or by the layperson. While we expect that the more explicit and precise real world semantics of such an ontology would positively affect the design and implementation of software systems in terms of their level of adaptability to change, at this stage of the research it must be noted that further empirical work is necessary in order to ascertain benefits and limitations of such a representation. In terms of future work we intend to progress along the following directions:

- First, refine the representation of *ContractSpecifications* and its parts, specifically those parts of contracts, which model the timeline that the parties must abide by, generating a plan of commitments which can be used during the course of the contract's execution to determine whether the parties are respecting their obligations or alternatively if different courses of action must take place. This would give rise to various new states (possibly of the parties or even also of the contract specification) related to the successful completion or breach of the contract.
- Second, model a typology of contract types (e.g., sale, rental, etc.) and respective commitment types (for example, payments, provision of goods/services, periodic commitments, on-demand commitments, etc.).
- Third, formalise the ontology in an ontology language such as the Web Ontology Language (OWL), populating the model with instance data and testing for consistency and correctness.
- Fourth, since laws and regulations impose constraints on contracts (in all their manifestations, e.g. specifications, stipulations and executions) it becomes necessary to investigate the ontological nature of such regulatory frameworks and their relationship with contracts. Such an analysis is required since the validity of contracts must be explicitly or implicitly consistent with the fulfillment of regulatory constraints.

References

1. Williamson, O.E.: Transaction Cost Economics: How it Works; Where it is Headed. De Economist 146(1), 23–58 (1998)
2. Reve, T.: The firm as a nexus of internal and external contracts. In: Foss, N.J. (ed.) The Theory of the Firm: Critical Perspectives on Business and Management, pp. 310–334. Routledge, London (2000)
3. Ayotte, K., Hansmann, H.: Economic and Legal Entities as Transferable Bundles of Contracts. Working paper (2010), `http://www.law.northwestern.edu/colloquium/law_economics/documents/Ayotte_Hansmann.pdf` (last accessed December 01, 2011)

4. Ijiri, Y.: Theory of Accounting Measurement. Studies in Accounting Research #10. American Accounting Association, Sarasota, Florida (1975)
5. McCarthy, W.E.: The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment. The Accounting Review 57(3), 554–578 (1982)
6. Geerts, G.L., McCarthy, W.E.: An ontological analysis of the economic primitives of the extended-REA enterprise information architecture. International Journal of Accounting Information Systems 3, 1–16 (2002)
7. Geerts, G.L., McCarthy, W.E.: Policy-Level Specifications in REA Enterprise Information Systems. Journal of Information Systems 20(2), 37–63 (2006)
8. Hruby, P., Kiehn, J., Scheller, C.: Model-Driven Design Using Business Patterns. Springer (2006)
9. Lévine, P., Pomerol, J.C.: From Business Modeling Based on the Semantics of Contracts to Knowledge Modeling and Management. In: Proceedings of the 34th Hawaii International Conference on System Sciences. IEEE (2001)
10. Lowe, E.J.: Ontology. In: Honderich, T. (ed.) The Oxford Companion to Philosophy, p. 634. Oxford University Press, New York (1995)
11. Gruber, T.R.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition 5(2), 199–220 (1993)
12. Smith, B.: Beyond Concepts: Ontology as Reality Representation. In: Varzi, A., Vieu, L. (eds.) International Conference on Formal Ontology and Information Systems (2004)
13. Searle, J.R.: The Construction of Social Reality. Free Press (1997)
14. Sider, T.: Four-Dimensionalism: An Ontology of Persistence and Time. Oxford University Press, USA (2002)
15. Partridge, C.: Business Objects: Re-Engineering for Re-Use. Butterworth-Heinemann (1996)
16. IDEAS Group: The IDEAS Model, `http://www.ideasgroup.org/foundation/`
17. Kabilan, V.: Using Multi-tier Contract Ontology to Model Contract Workflow Models. Ph.D. Thesis, Stockholm University, Sweden (2003), `http://kth.diva-portal.org/smash/get/diva2:7604/FULLTEXT01`
18. Kripke, S.: Naming and Necessity. Harvard University Press, Cambridge (1980)
19. Lewis, D.: On the Plurality of Worlds. Blackwell (1986)
20. Daskalopulu, A., Sergot, M.: The Representation of Legal Contracts. AI and Society 11(1/2), 6–17 (1997)
21. Chopra, A.K., Oren, N., Modgil, S., Desai, N., Miles, S., Luck, M., Singh, M.P.: Analyzing Contract Robustness through a Model of Commitments. In: Weyns, D., Gleizes, M.-P. (eds.) AOSE 2010. LNCS, vol. 6788, pp. 17–36. Springer, Heidelberg (2011)
22. Ferrario, R., Janiesch, C., Guarino, N., Oberle, D., Probst, F.: Towards an Ontological Foundation of Services Science. The General Service Model. In: The Proceedings of Wirtschaftsinformatik, Zurich, Switzerland (2011)
23. Ferrario, R., Guarino, N., Fernandez Barrera, M.E.: Towards an Ontological Foundations for Services Science: the Legal Perspective. In: Sartor, G., Casanovas, P., Biasiotti, M., Fernandez Barrera, M. (eds.) Approaches to Legal Ontologies. Law, Governance and Technology Series, vol. 1, pp. 235–258. Springer (2011)

A Proposal of an Event Ontology for Urban Crowd Profiling

Mizar Luca Federici[1], Andrea Gorrini[2], Lorenza Manenti[3], and Fabio Sartori[3]*

[1] CROWDYXITY s.r.l. - Crowd Dynamics and Complexity
Via Ventura 3, 20134 - Milano Italy
m.federici@crowdyxity.com

[2] INFORMATION SOCIETY Ph.D. Program
University of Milan-Bicocca
Via Bicocca degli Arcimboldi 8, 20126 - Milano Italy
a.gorrini@campus.unimib.it

[3] CSAI - Complex Systems and Artificial Intelligence Research Center
University of Milan - Bicocca
Viale Sarca, 336, 20126 - Milano Italy
{manenti,sartori}@disco.unimib.it

Abstract. The definitions of "event" and "crowd" are still representing controversial issues that have been tackled by different disciplines like Sociology, Philosophy and Computer Science. The proposed ontology of events takes advantage of results and perspectives already present in literature and in available resources, like DBpedia. Events, such as celebrations, concerts, sport matches and so on, are, in this work, defined as structured entities spatially and temporally confined, codified by a specific script, and participated by urban crowds. The integration of an ontology of "event" with an ontology of "crowd" constitutes the originality of this work. A conceptual framework has been defined, and then implemented in Protègè, to create a versatile tool to profile crowds. In the paper the assumptions that underline the development of the ontology are introduced, then its implementation in Protègè and its application to a case study is presented.

1 Introduction

There is scarce agreement on what an event is and how it should be considered in the systematization of an ontology. From a philosophical point of view, realists consider that events are real things and that they should be considered to belong to the same class of objects. This position is adversed by non-realists that neglect that events can provide a fixed framework of reference for an ontology that could give account of our practices of definition. Also, while objects are said to exist in clear temporal and spatial boundaries, events instead can be said to occur, or to take place, but their boundaries of existence are not clearly defined. The

* Authors are listed in alphabetical order

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 97–104, 2012.

dichotomy "event" vs "object" is not the only possible one and philosophical positions at this respect are several [1].

Gero Mühl and Fiege [2] introduced the notion of time and space in the event definition: a concrete experience in the world can be described as a sequence of events happening over the space and time. An event instance is a concrete object presenting attributes that describe it. According to this view, events can be seen as the combination of other events, which participate in the definition of a more complex one. The last definition is very similar to the one given by Kaneiwa et al. [3], who started from the consideration that objects and properties necessary to describe actions and changes in the real world are static, while events are typically dynamic. In classical approaches, events were methods for classifying relevant patterns of modifications rather than concrete entities of the world [4], as well as changes occurring in the discrete phases of a process [5]. Taking care of the object-oriented nature of events suggested by Guarino et al. [6], or frameworks like SUMO [7] and OpenCyc[1], Kaneiwa et al. [3] proposed an upper ontology for event description, where they tried to classify the types of events (e.g. natural events, artificial events). Then, each event is specified according to a couple of ontological views: *component structures* and *semantic functions* of it. Finally, they introduced event relations, like *causal* and *next-event* relations to point out the differences between instances and classes of events.The main difference between all the approaches introduced so far and the approach here presented is that event classification is strictly related to the kind of crowd that participates it. In other words, no definition of event is possible if it cannot be verified by a group of people sharing a given set of attributes at a given time and space.

The organization of big events[2] (such as trade exhibition, musical, artistic and cultural festivals) is becoming a consolidated urban policy to promote and refurbish urban areas (in particular dismissed industrial areas). These kinds of intervention are aimed at enhancing the city potentiality, catalyzing investments, improving urban services, creating a sense of belonging of citizens to the city [8]. The scarcely predictable impact of the extraordinary touristic flows during the event will make difficult for organizers and authorities to plan and manage the events. In particular, large cities have to be prepared to avoid disruption, and to guarantee accessibility and security. We consider as urban crowds the participants to this kind of events.

If the definition of what an event is remains an open issue, no agreement can be found in literature about what a crowd is either, because of the difficulty in empirical investigation of the phenomenon. Early interest in studying crowd started by the pioneering study of Gustave Le Bon [9], who defined crowd behaviour as irrational and a potential threat to society. Far from this perspective, the ESIM-Elaborated Social Identity Model [10] proposes a social-normative conception of collective behavior, arguing that social norms continue to shape behavior of people in the crowd. Taking advantage from these assumptions, the

[1] www.opencyc.org

[2] Properly named as "festivalization of the city".

most accepted definition of what a crowd is cites: “A crowd can be defined as a gathering of 20 people (at least), standing in close proximity at a specific location to observe a specific event, who feel united by a common social identity, and who are able to act in a socially coherent way, despite being strangers in an ambiguous or unfamiliar situation” [11, p. 43]. Considering the variability of crowd size and typology, Elias Canetti [12] codified a very detailed classification of crowds on the basis of criteria such as attitude to grow, attributes of density and equality, nature of the goal. Applying this grid, Canetti identified a variety of crowds: open and closed, stagnating and rhythmic, quick and slow. Canetti identified with the term “discharge” the cause of the assembling of the crowd: discharge is a mechanism that transforms single individuals in a proper crowd where individual differences are dropped.

An ontology that combines events and crowds offers the possibility to collect and systematize knowledge of these important phenomena that will require a great attention in the immediate future. With this work we want to propose a computational framework for crowd profiling and simulation, in order to support decision makers, designers and organizers of big events. In relation to several case studies performed by CSAI Research Centre and CROWDYXITY s.r.l., a spin-off of the University of Milan-Bicocca, the further aim of the framework is to organize and store gathered data related to crowd and pedestrian dynamics in high density situation during the events, such as: pedestrian walking speed, queue formations, waiting time, level of service, and so on.

After the previous theoretical introduction, in Section 2 a full description of our event ontology, both from a conceptual and a technical point of view, will follow. At last in Section 3, we will show a case study related to an urban crowd at a musical event. Final remarks and conclusions will be presented in Section 4.

2 The Ontology of Events: Definition and Implementation

In this section, starting from the previous discussion about the term “*event*”, and considering the state of the art in the ontology literature, we will introduce our point of view on the definition of events and crowds, underlining their relationship. Then, we will present the ontology implementation by means of Protègè platform: a free, open source ontology editor and knowledge-based framework.

Starting from literature, we introduce the definition of an event as a structured entity, spatially-temporally defined. We enlarge this basic definition in order to consider aspects that are primary elements in the development of a computational model for crowd profiling and simulation, that is the target of this ontology.

The first investigated aspect is related to the spatial extension of events: every event should be located into a specific *place* (e.g. a building, a station, a park and so on), composed of *venue*, *entrances* and *exits*, and several *utilities* (i.e., objects that are necessary to support the spatial structure of the event). All of them are primary elements that must be considered in modeling the crowd behavior in the environment.

The second aspect is related to the temporal *duration* of events, that is composed of starting time, execution time, and ending time. From the participants point of view, the latter corresponds to three main time phases: *inflow*, *involvement*, and *downflow*.

We propose a more detailed definition of *persons* who take part in the event, basically conceptualized as audience, taking into account also other subcategories referred to the organizers of activities: *staff*, *security* and *artist*.

In the description of the architecture of the events, we point out that every event is characterized by a *script*: *"A script, as we use it, is a structure that describes an appropriate sequence of events in a particular context. [..] For our purposes, a script is a predetermined, stereotyped sequence of actions that define a well-known situation."* [13, p. 151]. People who participate the event as organizers are supported in their activities by specific scripts, which represent predefined sequences of procedures and tasks. All the organizers, depending on the level of intensity of their performance, concur to the growth of the *discharge*. As a consequence, people who participate the event as audience can live the discharge, depending on their level of motivation. Taking advantage from the Canetti's Theory, the discharge produced by organizers is able to assemble people within a *crowd*, that can be of different types (as previously defined in Sec. 1).

At the end, we propose a structural definition of event, as composed of subevents which are modeled following the same categories as the native event. The latter is a strategy that allows to represent complex events composed of subevent (such as big event participated by urban crowds), by means of the unique conceptual model (see Sec. 3). In Figure 1 we depict a synthetical representation of the ontology of events as proposed in this work.

After the conceptual analysis, we focus on the development of the ontology by means of Protègè[3] platform, the standard *de facto* editor for ontologies. We point out the following concepts: *Event* is a spatially-temporally structured entity, participated by person, and characterized by script; *Place* is the spatial extension of an event; *Venue* is the space where the event takes place; *Entrance*, those are the gates which permit persons to access the venue; *Exit*, those are the gates which allow persons to leave the venue; *Utility*, those are necessary objects to support the spatial structure of the event; *Duration* is the temporal duration of an event; *Inflow* is the starting time of the event; *Involvement* is the execution time of the event; *Downflow* is the ending time of the event; *Person*, those are the event participants; *Staff*, those the event managers; *Security*, those are the security managers; *Artist* is the manager of the event performance; *Script* is the procedural structure that characterizes an event; *Discharge* is the product of the performance, able to assemble people within a crowd; *Crowd* is a gathering of people, standing in close proximity at a specific location to observe a specific event, who feel united by a common social identity.

The aim of the development is to adequate the proposed event ontology to the existing works in literature. For this reason, we adopted the concepts of *Place*

[3] http://protege.stanford.edu/

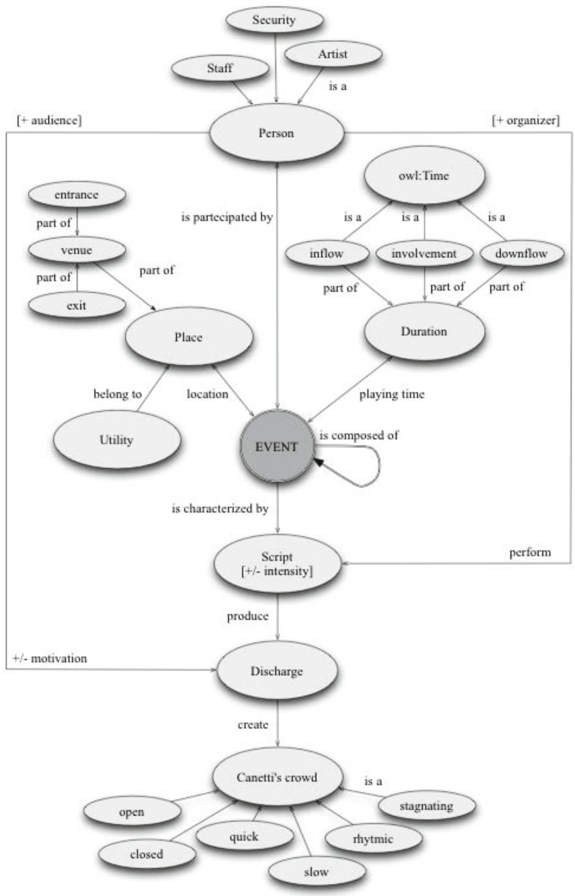


Fig. 1. A schematic representation of the concepts and relationships in our Event Ontology

and *Person* as they are defined in DBpedia Ontology[4]. The concepts of *Inflow*, *Involvement*, *Downflow* are modelled by means of the OWL-Time ontology[5]. We use a previous crowd classification ontology, developed within the research activities of CSAI [14] to implement concepts about *Crowd*.

Starting from these assumptions, the implementation of our Event Ontology in Protègè wass organized into two main phases: *Loading of existing ontologies*, in order to extend previous results found in literature. We imported in Protègè the version 3.6 of DBpedia Ontology, and the latest version of OWL-Time Ontology. Moreover, in order to exploit the previous work of conceptualization of the theories on crowd, we imported the Canetti's Crowd Ontology; *Classes, proper-*

[4] http://dbpedia.org/ontology/

[5] http://www.w3.org/TR/owl-time/

Table 1. Metadata representation of properties and datatypes for Event Ontology. Note that *p1* is the prefix for DBpedia ontology.

Label	**Domain**	**Range**	**Comment**
location	*Event*	*p1:Place*	relates an event to its spatial location
playingTime	*Event*	*Duration*	relates an event to its temporal duration
belongTo	*Utility*	*p1:Place*	relates an utility to its spatial position
isPartecipatedBy	*Event*	*p1:Person*	relates an event to involved people
isComposedOf	*Event*	*Event*	relates an event to its sub-events
isCharacterizedBy	*Event*	*Script*	relates an event to its script
produce	*Script*	*Discharge*	relates a script to the discharge produced
perform	*p1:Person*	*Script*	relates a person to the performed script
motivation	*p1:Person*	*Discharge*	relates a person to the experienced discharge
create	*Discharge*	*Crowd*	relates a discharge to the created crowd
motivationLevel	*p1:Person*	*xsd:integer*	level of motivation in participating to a discharge
intensity	*Script*	*xsd:integer*	level of intensity of the performed script
role	*p1:Person*	*xsd:String*	role of a person respect to the event

ties and data-types definition, where the innovative aspects of this Ontology of Events have been implemented.

An overview on the metadata description of properties and data-types are presented in Table 1, in which label, domain, range such as a textual definition of these elements are shown.

The ontology has been developed in Protègè, adopting the version 3.4 of the platform for its user-friendly interface and the integration with Jambalaya[6] plugin to visualize the knowledge bases the user has created.

3 Case Study: The Concert

In this section we will propose to apply our Event Ontology to describe and profile the urban crowd that participates to a concert. We chose this specific scenario as case study due to its ability to represent a typical urban big event. The chosen study represents also an opportunity to explain how a complex event can be divided into sub-events. Moreover, thanks to several research activities performed in this field by CSAI and CROWDYXITY s.r.l., we hold a deep knowledge about this kind of event and its organization, and we have collected several empirical data.

In particular, we refer to the observation of the *Jovanotti Ora Tour*[7], performed at the Mediolanum Forum, in May 11^{th}, 2011 - Assago, Milano (Italy). In the following we propose the analysis of the whole event *"Concert"*, and its sub-events *"Song"*.

We allocated the model as below: *Event* is the "Jovanotti Ora Tour", 05/11/2011; *Place* is the "Mediolanum Forum - Assago", Milano (Italy); *Venue* is the standing area and the tribunes of the Mediolanum Forum; *Entrance* is one

[6] http://www.thechiselgroup.org/jambalaya
[7] http://www.soleluna.com/

entrance, positioned in front of the stage; *Exit*, those are two exits positioned in front of the stage; *Utility*, those are the stage, sound-lights mixer, barriers, and so on; *Duration* is May 11^{th} , 2011; *Inflow* is from 8:00 pm to 9:30 pm; *Involvement* is from 9:30 pm to 11:30 pm; *Downflow* is 11:30 pm to 12:00 pm; *Person*, who participates to the Jovanotti Ora Tour, 05/11/2011; *Staff* is the Jovanotti Ora Tour Staff; *Security*, those are stewards, firefighters, paramedics; *Artist* is Jovanotti (and his band); *Script*, those are procedures to perform the concert from both the artistic and technical point of view, ensuring security conditions; *Discharge*, those are, depending on the intensity of the performance script, the track list, lights and sound effects; *Crowd* is closed, quick and rhythmic crowd[8].

Starting from the event description as a structured entity composed of 15 sub-events *"Song"*, the chosen case study can be analytically defined as a sequence of songs (i.e. the tack list of the concert). Each event *"Song"q* represents the atomic part of the main event *"Concert"*. The main differences with respect to the native event are related to the temporal duration and the script. The involvement starts at May 11, 2011 from 10:10pm to 10:14pm (no in-flow and down-flow phases are defined). The script overlaps with musical composition of the song, for both the speech and the melody. Due to its atomic definition, the event *"Song"* cannot be divided into sub-events.

4 Final Remarks and Conclusions

This work is aimed at supporting decision makers, designers and organizers in the managing of big events, by means of a computational framework for crowd profiling and simulation. The knowledge of crowd could be a useful contribution for a successful management of each phase of an event. We refer in particular to the best practices related to ensure security during the event. To achieve this objective we have illustrated a synthetic theoretical discussion about events and crowds. The implementation of this framework in the Protègè platform was developed thanks to the integration with existing ontologies such as DBpedia and OWL-Time. The innovative perspective of this work is based on the relationship between events and crowds: this fusion offers the possibility to collect and systematize knowledge, data and information on big event that will require a great attention in the immediate future.

Future works are devoted to fully integrate the "Event" and "Crowd" ontologies to make them capable to describe big urban events and the urban crowds who populate them, with the final aim to build up a complete and general conceptual and computational framework for supporting organizers in forecasting possible critical situations. A possible application scenario will be the upcoming World Meeting of Families in Milan[9].

[8] The profiling of crowd is based on the Canetti's Theory and on several physical characteristics of the environment and people within the crowd (e.g. density, lifespan, growth and so on).

[9] http://www.family2012.com/

Acknowledgement. We would like to thank Barbara Diana, Ph.D. student of the University of Milan–Bicocca, for the contribute on urban big event from her master thesis degree.

References

1. Casati, R., Varzi, A.: Fifty years of events: An annotated bibliography 1947 to 1997. Philosophy Documentation Center, Bowling Green, OH (1997)
2. Mühl, G., Fiege, L., Pietzuch, P.R.: Distributed event-based systems. Springer (2006)
3. Kaneiwa, K., Iwazume, M., Fukuda, K.: An Upper Ontology for Event Classifications and Relations. In: Orgun, M.A., Thornton, J. (eds.) AI 2007. LNCS (LNAI), vol. 4830, pp. 394–403. Springer, Heidelberg (2007)
4. Allen, J.F., Ferguson, G.: Actions and events in interval temporal logic. J. Log. Comput. 4(5), 531–579 (1994)
5. Sowa, J.F.: Knowledge representation: logical, philosophical and computational foundations. Brooks/Cole Publishing Co., Pacific Grove (2000)
6. Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Schneider, L.: Sweetening Ontologies with DOLCE. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS (LNAI), vol. 2473, pp. 166–181. Springer, Heidelberg (2002)
7. Niles, I., Pease, A.: Towards a standard upper ontology. In: FOIS, pp. 2–9 (2001)
8. Krantz, M., Schätzl, L.: Marketing the city. European Cities in Competition, 468–491 (1997)
9. Le Bon, G.: The crowd (1895). Viking, New York (1960)
10. Reicher, S.: Crowd behaviour as social action. Rediscovering the Social Group: A Self-Categorization Theory, 171–202 (1987)
11. Challenger, R., Clegg, C., Robinson, M.: Understanding Crowd Behaviours, vol. 1. Cabinet Office (2009)
12. Canetti, E., Stewart, C.: Crowds and power. Farrar, Straus and Giroux (1984)
13. Schank, R., Abelson, R., University, Y.: Scripts, plans, and knowledge. Yale University (1975)
14. Bandini, S., Manenti, L., Manzoni, S., Sartori, F.: A knowledge-based approach to crowd classification. In: Peacock, R.D., Kuligowski, E.D., Averill, J.D. (eds.) Pedestrian and Evacuation Dynamics, pp. 515–525. Springer, US (2011)

Ontology Evolution with Semantic Wikis

Mauro Dragoni and Chiara Ghidini

FBK-irst, Via Sommarive 18 Povo, I-38123,Trento, Italy
[dragoni,ghidini]@fbk.eu

Abstract. One of the challenges of using ontology evolution approaches is the capability of exposing the ontology with information that may be used by third-party tools for tracking the updates carried out on the ontologies. In this paper we present and enhanced version of the MoKi tool equipped with an ontology evolution approach that permits to evolve an ontology by providing a mechanism for facing the tracking challenge. By considering, as use case, the context of the Organic.Lingua EU-project, we will discuss the effectiveness of the proposed approach and possible drawbacks.

1 Introduction

Ontologies are dynamic entities that evolve over time because they are affected by the necessity of applying changes in the domain, in the conceptualization, or in their specification. As stated in [25], the ontology evolution may be defined as "the timely adaptation of an ontology to the arisen changes and the consistent propagation of these changes to dependent artifacts."

The management of ontology evolution has several challenges associated, ranging from the adequate control of ontology changes to the administration of ontology versions. Ontologies evolution refers to the activity of facilitating the modification of an ontology by preserving its consistency; it can be seen as a consequence of different activities during the development and, mainly, the maintenance of the ontology.

Starting from a high level perspective, we can classify ontology evolution in two main scenarios: the first one is that only one ontology is created, maintained, and made evolve, by one or more users; the second, and more complex one is that different users work on different ontologies, and then, these ontologies are merged, and made evolve, in one single inter-ontology. The challenges raised by using an ontology evolution mechanism become more complex in scenarios where concepts defined in the ontology are used to tag resources that are then retrieved by using third-party tools or web search environments. By taking into account this scenario, the tool used for managing the ontology and its evolution has to provide the capability of injecting, into the ontology, some information for maintaining the retrieval effectiveness of the search environments used to retrieve resources that are tagged with concepts that have been updated or deleted.

In this paper we analyze the scenario explained above by considering, as use case, the context of the Organic.Lingua EU-project[1]. This project aims to create a multilingual portal where users are able to retrieve agricultural resources tagged with concepts defined into an ontology. The ontology is managed by using a semantic wiki tool, called

[1] http://www.organic-lingua.eu

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 105–116, 2012.

MoKi, that has been equipped with an ontology evolution approach focused on the maintenance of the retrieval effectiveness of the Organic.Lingua platform.

The paper is organized as follow: Section 2 presents a review on the ontology evolution field. Section 3 presents the MoKi tool, while in Section 4 we discuss how MoKi will be used for facing the ontology evolution challenge and we compare the presented enhanced version of MoKi against the other semantic wikis presented in the literature. Section 5 shows which are the issues raised when ontology evolution approaches are used; finally, Section 6 concludes.

2 Related Work

Several approaches have been presented in the literature about ontology evolution. The aspects that are taken into account may be split in two different categories:

- **Schema evolution:** it is the ability to change the schema of the ontology without loss of data and by maintaining the consistency of the ontology, moreover, it has to be possible to access to both old and new data through the new ontology schema;
- **Schema versioning:** it is the ability to access all the data (both old and new) through all versions of the ontology. A version is a reference that labels a quiet point in the definition of a schema. Therefore, all resources have to be retrievable by using coherent concepts and by using every historic definition of the ontology schema.

Both aspects are strictly correlated with the context of the Organic.Lingua project due to the necessity of maintaining the retrieval effectiveness of the system through the changes that are carried out on the ontology.

Schema Evolution. The evolution of ontology schema is generally composed of two phases: the planning of the changes, and their implementation.

The planning of the changes consists of identifying the potential consequences of a change, and estimating what needs to be modified to accomplish a change. Such analysis, that derives from the software engineering environment, is very helpful to estimate the cost and effort required to implement the requested change. In [17] the author describes in detail how the change of a concept in the ontology might imply a cascade of changes (propagation) that may affect the entire ontology.

By analyzing the propagation of changes, it is possible to estimate which is the cost of the ontology evolution. As stated in [17], the cost of evolution is a key element in the decision on propagating changes through the ontology or not, and in [23] the authors provide an overview on approaches related to the estimation of the cost of evolution and propose a parametric model for the estimation of costs to build, maintain, and evolve and ontology.

The second phase is related to the implementation of the changes. In the literature there are identified four different kinds of approach used to propagate the effects of changes implemented in an ontology:

- immediate conversion (or coercion) [12] [15]
- deferred conversion (or screening) [1] [20]

– explicit deletion [3]
– filtering [1] [20]

After the implementation of the changes, it is sometimes necessary to restructure the ontology in order to maintain the same information capacity [14] defined as the semantic equivalence between different versions of the ontology [19]. This step may introduce inconsistencies in the ontology schema that have to be managed.

There are two different schools of thought about how to face this problem: the first one is the "consistency maintenance", that it is a conservative approach in which the system is kept consistent at all costs. Some approaches related to consistency maintenance are presented in [25] and [11]. While, the second strategy simply consists in the "inconsistency management", in which inconsistencies are considered inevitable and, therefore, it is necessary to manage them. The main studies in the literature about inconsistency management are related to the localization of inconsistencies, for example [6] it is presented an approach based on the use of sub-ontologies to identify inconsistencies, while in [18] the authors present a logic-based method to detect some kinds of inconsistencies.

Schema Versioning. Versioning is in general the mechanism that allow users to keep track of all changes applied to the definition of something, and to undo changes by rolling back to previous versions. The same principle is applied to ontologies, in which it is necessary to track the changes applied to the ontology schema, in order to permit to users to roll-back to previous versions if it is necessary.

There are two main strategies that are adopted to establish the version of an ontology: "state-based" and "change-based". The "state-based" versioning consists in considering the state of the ontology at certain moment in time; a new state is created each time that a change is applied by the system to the ontology schema. An example of system that supports such a versioning strategy is described in [8].

The second way to manage schema versioning is a "change-based" approach (or "operation-based") that consisting in storing information about the precise changes or explicit operations that are performed on the ontology. The advantages of this approach, with respect to the previous one, is that it is simpler to compare different versions of an ontology and to implement undo/redo mechanisms. An example of system implementing a change-based approach is proposed in [13].

3 MoKi Tool

MoKi[2] is a collaborative MediaWiki-based [9] tool for modeling ontological and procedural knowledge. The main idea behind MoKi is to associate a wiki page, containing both unstructured and structured information, to each entity of the ontology and process model. From a high level perspective, the main features of MoKi[3] are:

[2] See http://moki.fbk.eu

[3] A comprehensive description of MoKi can be found in [4].

- the capability to model different types of conceptual models in an integrated manner. In particular the current version of MoKi is tailored to the integrated modeling of ontological and procedural knowledge;
- the capability to support on-line collaboration between members of the modeling team, including collaboration between domain experts and knowledge engineers.

In the context of the Organic.Lingua project, the use of MoKi has been focused on the modeling of the ontological knowledge only, while the collaborative feature is useful due to the structure of the modeling team that is composed by heterogeneous groups of domain experts and knowledge engineers situated in different geographical regions.

The following subsection illustrates how these features are realized in the generic MoKi architecture.

Modeling integrated ontological and procedural knowledge The capability of modeling integrated ontological and procedural knowledge is based on different characteristics of MoKi. MoKi associates a wiki page to each *concept*, *property*, and *individual* in the ontology, and to each (complex or atomic) *process* in the process model. Special pages enable to visualize (edit) the ontology and process models organized according to the generalization and the aggregation/decomposition dimensions respectively. The ontological entities are described in Web Ontology Language (OWL [24]), while the process entities are described in Business Process Modeling Notation (BPMN [16]).

Supporting collaboration between domain experts and knowledge engineers MoKi is an on-line tool based on MediaWiki, thus inheriting all the collaborative features provided by it. In addition MoKi facilitates the collaboration between domain experts and knowledge engineers by providing different access modes to the elements described on the model, as illustrated in Figure 1 for the ontology concept "Mountain".

MoKi allows to store both *unstructured* and *structured* descriptions of the elements of the models, as shown on the left hand side of Figure 1. The unstructured part contains a rich and often exhaustive description of knowledge better suited to humans, usually provided with linguistic and pictorial instruments. Instead, the structured part is the one which is used to provide the portion of knowledge which will be directly encoded in the modeling language used to describe the specific element (OWL in the case of the concept "Mountain"). The advantage of storing the unstructured and structured descriptions in MoKi is twofold. First, informal descriptions are usually used to provide the initial description upon which the formal model is built, and to document the elements of the model (e.g., for future access and revisions). Storing the unstructured and structured descriptions in the same tool can facilitate the interplay between these parts. Second, domain experts, who usually create, describe, and review knowledge at a rather informal/human intelligible level, may find the unstructured part their preferred portion of page where to describe knowledge, while knowledge engineers should be mainly focused on the descriptions contained in the structured part. Nevertheless, by using the same tool and accessing the same pages, all of them can be notified of what the others are focused at. Moreover, the discussion facilities of wikis, together with special fields for comments, can be used by both roles to discuss on specific parts of the model.

The reader, may found more details about the general features of MoKi in [5].

Fig. 1. Multi-mode access to a wiki page

4 Ontology Evolution with MoKi

By considering the aims of the Organic.Lingua project, MoKi implements an evolution mechanism that permits to achieve the following goals:

- to modify the definition of a concept and at the same time to maintain an association of deprecation with the old definition;
- to delete a concept and at the same time to track the changes;
- to grant retrieval effectiveness after an update or a deletion of a concept.

The main issue related to the ontology evolution in the Organic.Lingua project is to maintain the retrieval effectiveness of the platform when concepts are both updated and/or deleted. In this Section we present how these goals are reached with the use of MoKi in the context of the Organic.Lingua project.

Concept Update. The concept update is intended as an action performed by a user consisting on modifying the definition with which a concept is identified in the ontology. A concept may be updated in different ways:

- the concept definition is only modified: assuming to have a concept defined as “A”, it is then renamed as concept “B”;
- the concept is split in two or more concepts: assuming to have in the ontology a concept defined as “A”, it may be split in the set of concepts “B”, “C”, and “D”.

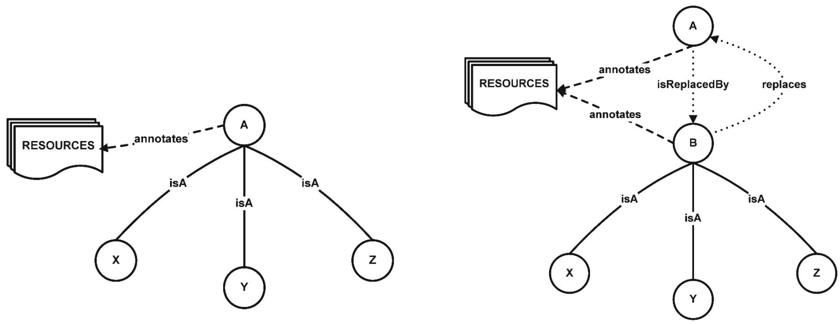


Fig. 2. Concept update by modifying its definition

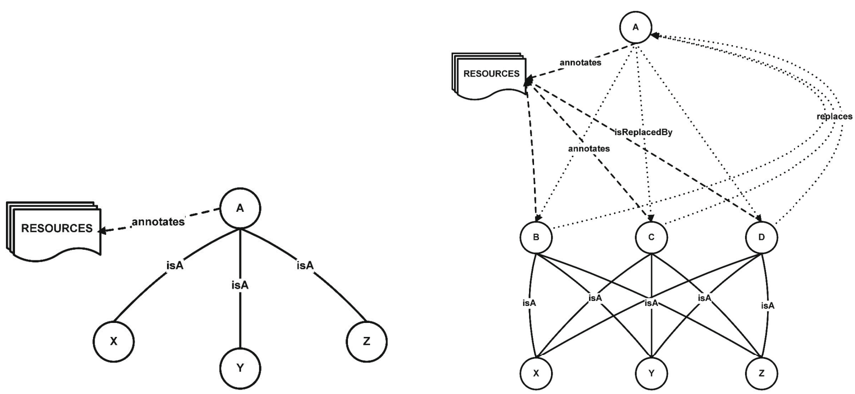


Fig. 3. Concept update by splitting the original concept in two or more new concepts

In this context, it is desirable that in both cases the old definition of the concept "A" is maintained in the ontology, because, as explained earlier, it is necessary that the resources stored in the repository, that has been annotated with the concept "A", still remain retrievable when users look for resources annotated with the concept "A".

Figures 2 and 3 present the two scenarios respectively when a concept is updated by modifying only its definition, and when a concept is split in two or more concepts.

The left parts of the images show the ontology situation before the update, while the right parts show how the ontology evolves after the two possible concept updates. In both cases, the starting point is the scenario in which a concept subsumes ("isA" relationship) other concepts, (the same strategies may also be applied for different type of relations), and it is used to annotated a set of resources stored in the repository.

When a concept is updated (Figure 2), the ontology is modified by inserting a new concept that replaces the old one; however, the old one is maintained in the ontology. The associations between the old concept and its subsumed concepts are moved, and they are placed as subsumptions of the new concept. Two relations are created between the old concept and the new one: "isReplacedBy" and "replaces". The relations "isReplacedBy" and "replaces" are used for retrieval purposes because they permit to navigate through the old concept definitions that are maintained in the ontology in order to preserve the retrieval effectiveness. In fact, the resources annotated with the definition of the old concepts still remain retrievable when users perform queries containing the definition of the old concepts. Moreover, in order to preserve also the efficiency of the platform, it is possible to annotate the resources, previously annotated with the definition of the old concept, with the definition of the new one. This way, the resources are retrievable by using only the definition of the new concept; otherwise, it would be necessary to perform the retrieval operation in different steps.

The second case is the split of the original concept in two or more concepts (Figure 3). In this case, the operations described in the previous case, are repeated for the all concepts that have been created after the split of the original one. Therefore:

- the subsumptions relationships are copied for all new concepts;
- the couple of relations "isReplacedBy" and "replaces" are created for each new concept;
- the actions to preserve the effectiveness and the efficiency of the platform are performed.

Concept Deletion. Similarly to the concept update, the concept deletion has to be managed differently based on the relationships between the deleted concept and the other ones. Two different scenarios are expected: (i) the concept is a middle node of the ontology; and, (ii) the concept is a leaf of the ontology.

The case in which the concept is at the top of the ontology is not managed because it is supposed that the top concept is never been deleted.

Figures 4 and 5 show the two scenarios respectively when the deleted concept is a middle node or a leaf of the ontology.

In the first case, it is supposed that the concept "B" (middle node) is deleted, while in the second case, it is supposed that the deleted concept is "Y" (leaf node). It is desirable that in both cases the concepts "B" and "Y" are somehow maintained in the ontology, because, as explained earlier, it is necessary that the resources stored in the repository, that has been annotated with the concept "B" (or "Y"), still remain retrievable when users look for resources annotated with the concept "B" (or "Y").

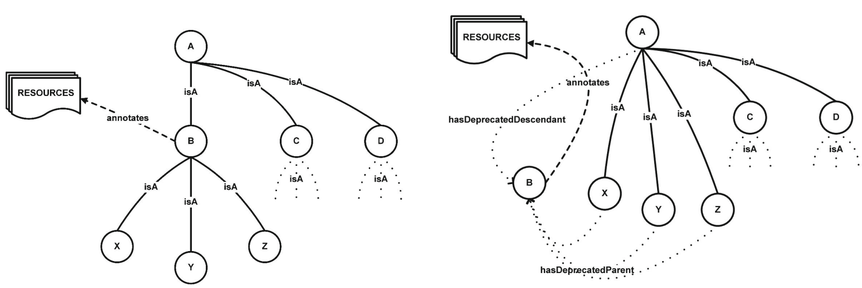


Fig. 4. Deletion of a middle-node concept

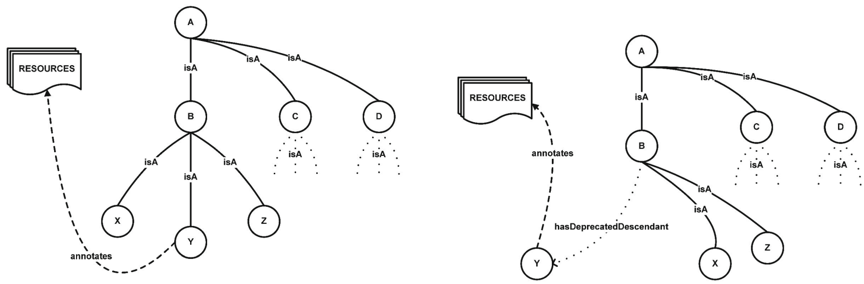


Fig. 5. Deletion of a leaf concept

The left parts of the images show the ontology situation before the deletion, while the right parts show how the ontology evolves after the two possible concept deletions. In both cases, the relation used to associated the concepts is the "isA" relationship, however, the same strategies may also be applied for different type of relations. It is also supposed that the resources are annotated by using only the deleted concept.

When the deleted concept is a middle node (the concept "B" in Figure 4), the subsumed concepts (in this case the concepts "X", "Y", and "Z") are directly associated with the parent concepts of "B" in order to preserve the consistency of the taxonomy. However, there is the possibility that a set of resources is annotated by using only the concept "B". To make these resources retrievable it has been decided to use the "hasDeprecatedParent" and "hasDeprecatedDescendant" relationships. The "hasDeprecatedParent" relationship associates the concepts that was descendants of the deleted concept, with the deleted concept itself. This way, when users perform queries containing the concept "X" and/or "Y" and/or "Z", the platform will also looks for resources that are annotated by using the concept "B". In the same way, when a user perform queries containing the concept "A", the "hasDeprecatedDescendant" relationship is exploited to retrieve the resources that have been annotated by using the concept "B"

In the second case (Figure 5), it is supposed that the deleted concept "Y" is a leaf node of the ontology. In this case, the evolution of the ontology is more simpler than the previous case. The "hasDeprecatedDescendant" relationship is created between the concepts "B" and "Y", and this relationship is exploited in the same way explained for the previous case. Therefore, when users look for resources annotated with the concept "Y", these resources still remain retrievable, while, when users look for resources annotated with the concept "B", the resources annotated with the concept "Y" will be retrieved too.

4.1 Ontology Evolution with Semantic Wikis: A Comparison

In the literature, wiki systems and semantic wikis have been mainly applied to support collaborative creation and sharing of ontological knowledge.

AceWiki [10] was developed in the context of logic verbalization, that is, the effort to verbalize formal logic statements into English statements and vice-versa. AceWiki is based on Attempto Controlled English (ACE), which allows users expressing their

Table 1. Comparison of state-of-the-art modeling wikis

	OWL support	Evolution mechanism
AceWiki	X	
SMW+	X	
IkeWiki	X	
OntoWiki	X	X
MoKi v.2	X	X

knowledge in near natural language (i.e. natural language with some restrictions). *Semantic MediaWiki+* [7], which includes the Halo Extension, is a further extension on Semantic MediaWiki with a focus on enhanced usability for semantic features. Especially, it supports the annotation of whole pages and parts of text, and offers "knowledge gardening" functionalities, that is maintenance scripts at the semantic level, with the aim to detect inconsistent annotations, near-duplicate entries etc.

IkeWiki [22] supports the semantic annotation of pages and semantic links between pages. Annotations are used for context-specific presentation of pages, advanced querying, consistency verification or drawing conclusions. *OntoWiki* [2] seems to focus slightly more directly on the creation of a semantic knowledge base, and offers widgets to edit/author single elements/pages and whole statements (subject, predicate, object).

We have compared the tools mentioned above, together with the versions of MoKi presented in this paper, against the ontology evolution features. The results are displayed in Table 1, where the columns refer to the capability of: (i) representing entities by using OWL syntax; and (ii) providing an ontology evolution algorithm when changes are carried out on pages.

As we can see from the table, all the compared wikis support the use of the OWL language for representing the entities, while only OntoWiki implements and ontology evolution approach. The approach implemented in OntoWiki is based on the EvoPat algorithm [21] that permits to define evolution patterns that may be applied to an ontology for evolving it. This strategy is merely related to the evolution of the ontology, but it does not take into account the necessity of tracking the historical aspect of the ontology. This way, with respect to the ontology evolution approach implemented in MoKi, an external tool is not able to infer which was the previous structure of the ontology.

5 Challenges on the Ontology Quality and Exposure

The use of ontology evolutions mechanisms needs further activities for checking the updated version of the ontologies that are generated as well as for verifying the impact of the carried out changes on the functionalities of third-party tools that, eventually, exploit the ontology for their activities (for instance, resource tagging). In this Section we want to highlight which are the challenges raised by using an ontology evolution mechanism and to verify if the approach implemented in MoKi faces them in an effective way.

5.1 Ontology Quality

The evolution of ontologies implies the possibility of introducing some mistakes into the ontology definition. For instance, after deletion operations, there is the probability that some concepts may remain orphans due to the removal of some relationships, or that there are individuals defined without using concepts. Some examples of imprecisions are: concepts and properties that do not have verbal descriptions, orphaned concepts, concepts that are not used to tag individuals, properties that do not have domain and/or range definitions, presence of non-shared concepts and/or properties, and individuals with no type defined.

In order to avoid the existence of these imprecisions, MoKi implements a service that permits to knowledge engineers to identify the elements that contains some errors.

This service checks the quality of the ontology by performing the following actions in order to avoid the issues listed above:

- Concepts checking: this check consists in verifying that every concept has a verbal description, that there are not orphaned concepts, concepts without individuals, and non-shared concepts.
- Individuals checking: this check consists in verifying that every individual has a type defined.
- Properties checking: this check consists in verifying that every property has a verbal description, that both domain and range are defined in each property, and that there are not non-shared properties.

Besides this automatic check, MoKi implements a further service that helps the knowledge engineer to discover ontology imprecisions. It consists in the use of questionnaires containing list of statements that have been automatically inferred from the domain model defined in the ontology used by the tool. Only statements formulated by using already existing complex concept expressions are displayed. A knowledge engineer analyzes the explanation of each inferred statement in order to understand how it has been inferred and if the statement is correct with respect to which should be the content of the ontology. This way, the knowledge engineer may exploit this service in order to identify possible imprecisions in the ontology and to adjust the ontology by providing/removing elements that cause the imprecisions.

5.2 Ontology Exposure

The challenge of exposing the ontology is a very important aspect not only in the context of the Organic.Lingua project.

Indeed, this challenge impacts on all scenarios in which there are tools that exploit ontologies produced and managed by third-party providers and that they use the concepts defined in the ontology for several activities, like the annotating one.

In these scenarios, the evolution of an ontology might be critical from the point of view of resource retrieval. Indeed, the external tools have to be able to understand how the changes are represented in the ontology in order to update the set of concepts available for the tagging activity, as well as, to provide a mechanism in the search environment for retrieving the resources tagged with the old versions of the ontology.

The current version of MoKi exposes the ontology in Linked Data format to external tools that use it for supporting the tagging activity of the resources that are deployed on the Organic.Lingua portal.

The approach implemented in MoKi that is described in Section 4, provides a set of relationships that permit to reconstruct the changes carried out on the ontology. This way, external tools may exploit these relationships for updating the interface provided to their users, while the search environment may use these relationships for maintaining its effectiveness.

6 Conclusions

In this paper we presented how the challenge of ontology evolution may be faced by using semantic wiki tools. In particular, the use case that we have considered takes into account the scenario in which an ontology is used for tagging resources available on the web. We described how the presented version of MoKi is able to manage the ontology evolution mechanism by providing a set of relationships that permit both to external tools and to search environment, to maintain the knowledge of the changes carried on the ontology.

Acknowledgments. Organic.Lingua is funded under the ICT Support Program of the EU Commission (Grant Agreement Number 270999).

References

1. Andany, J., Léonard, M., Palisser, C.: Management of schema evolution in databases. In: Lohman, G.M., Sernadas, A., Camps, R. (eds.) VLDB, pp. 161–170. Morgan Kaufmann (1991)
2. Auer, S., Dietzold, S., Riechert, T.: OntoWiki – A Tool for Social, Semantic Collaboration. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 736–749. Springer, Heidelberg (2006)
3. Banerjee, J., Kim, W., Kim, H.-J., Korth, H.F.: Semantics and implementation of schema evolution in object-oriented databases. In: Dayal, U., Traiger, I.L. (eds.) SIGMOD Conference, pp. 311–322. ACM Press (1987)
4. Ghidini, C., Rospocher, M., Serafini, L.: Moki: a wiki-based conceptual modeling tool. In: ISWC 2010 Posters & Demonstrations Track: Collected Abstracts. CEUR Workshop Proceedings (CEUR-WS.org), Shanghai, China, vol. 658, pp. 77–80 (2010)
5. Ghidini, C., Rospocher, M., Serafini, L.: Conceptual modeling in wikis: a reference architecture and a tool. In: The Fourth International Conference on Information, Process, and Knowledge Management, eKNOW 2012 (2012)
6. Haase, P., Stojanovic, L.: Consistent Evolution of Owl Ontologies. In: Gómez-Pérez, A., Euzenat, J. (eds.) ESWC 2005. LNCS, vol. 3532, pp. 182–197. Springer, Heidelberg (2005)
7. Hansch, D., Schnurr, H.-P.: Practical applications of semantic mediawiki in commercial environments - case study: semantic-based project management. In: 3rd European Semantic Technology Conference, ESTC 2009 (2009)
8. Klein, M.C.A., Fensel, D., Kiryakov, A., Ognyanov, D.: Ontology Versioning and Change Detection on the Web. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS (LNAI), vol. 2473, pp. 197–212. Springer, Heidelberg (2002)

9. Krotzsch, M., Vrandecic, D., Volkel, M.: Wikipedia and the semantic web - the missing links. In: Proc. of the 1st Int. Wikimedia Conference, Wikimania 2005 (2005)
10. Kuhn, T.: AceWiki: A Natural and Expressive Semantic Wiki. In: Proceedings of Semantic Web User Interaction at CHI 2008: Exploring HCI Challenges (2008)
11. De Leenheer, P., de Moor, A., Meersman, R.: Context dependency management in ontology engineering: A formal approach. J. Data Semantics 8, 26–56 (2007)
12. Lerner, B.S., Habermann, A.N.: Beyond schema evolution to database reorganization. In: OOPSLA/ECOOP, pp. 67–76 (1990)
13. Maedche, A., Motik, B., Stojanovic, L.: Managing multiple and distributed ontologies on the semantic web. VLDB J. 12(4), 286–302 (2003)
14. Miller, R.J., Ioannidis, Y.E., Ramakrishnan, R.: The use of information capacity in schema integration and translation. In: Proceedings of the 19th International Conference on Very Large Data Bases, VLDB (1993)
15. Nguyen, G.T., Rieu, D.: Schema evolution in object-oriented database systems. Data Knowl. Eng. 4(1), 43–67 (1989)
16. OMG. Business process modeling notation, v1.1, www.omg.org/spec/BPMN/1.1/PDF
17. Plessers, P.: An Approach to Web-Based Ontology Evolution. PhD thesis. Department of Computer Science, Vrije Universiteit Brussel, Brussel, Belgium (2006)
18. Plessers, P., De Troyer, O.: Resolving Inconsistencies in Evolving Ontologies. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 200–214. Springer, Heidelberg (2006)
19. Proper, H.A., Halpin, T.A.: Conceptual schema optimisation: Database optimisation before sliding down the waterfall. Technical Report 341. Department of Computer Science, University of Queensland, Australia (1998)
20. Ra, Y.-G., Rundensteiner, E.A.: A transparent schema-evolution system based on object-oriented view technology. IEEE Trans. Knowl. Data Eng. 9(4), 600–624 (1997)
21. Rieß, C., Heino, N., Tramp, S., Auer, S.: Evopat - pattern-based evolution and refactoring of rdf knowledge bases. In: International Semantic Web Conference (1), pp. 647–662 (2010)
22. Schaffert, S.: Ikewiki: A semantic wiki for collaborative knowledge management. In: 1st Int. Ws. on Semantic Technologies in Collaborative Applications, STICA 2006 (2006)
23. Simperl, E., Popov, I.O., Bürger, T.: ONTOCOM Revisited: Towards Accurate Cost Predictions for Ontology Development Projects. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 248–262. Springer, Heidelberg (2009)
24. Smith, M.K., Welty, C., McGuinness, D.L.: Owl web ontology language guide. W3C Recommendation, February 10 (2004)
25. Stojanovic, L.: Methods and Tools for Ontology Evolution. PhD thesis. University of Karlshrue, Karlshrue, Germany (2004)

Using Open Information Extraction and Linked Open Data towards Ontology Enrichment and Alignment

Antonis Koukourikos[1,2], Pythagoras Karampiperis[2], George Vouros[1], and Vangelis Karkaletsis[2]

[1] University of Piraeus, Department of Digital Systems. 80, Karaoli and Dimitriou Str, Piraeus, 18534
[2] Software and Knowledge Engineering Laboratory, Institute of Informatics and Telecommunications, National Center for Scientific Research "Demokritos"Agia Paraskevi Attikis, P.O.Box 60228, 15310 Athens, Greece

Abstract. The interlinking, maintenance and updating of different Linked Data repositories is steadily becoming a critical issue as the amount of published data increases. The wealth of information across the World Wide Web can be exploited in order to provide additional information about the way that an object is described in the real world. This paper proposes a method for discovering new concepts and examining the equivalence of properties in different LOD description schemas by using Open Information Extraction techniques on web resources. The method relies on constructing association graphs from the extracted information, proceeding to a transfer on the conceptual level using information previously known from the LOD repositories and examining the similarities and discrepancies between the produced graphs and the LOD descriptions, as well as between the graphs derived from different repositories.

Keywords: Ontology Enrichment, Ontology Alignment, Linked Open Data, Open Information Extraction.

1 Introduction

The need to describe in a consistent, machine-readable way the vast amount of information found on the Web has led to numerous initiatives related to the Semantic Web movement[1]. The Linked Data Initiative provides a set of guidelines and best practices for the publication and interlinking of different resources in RDF[2] format. The Linking Open Data community project[3] dedicates its efforts into collecting various datasets, publishing them as RDF triples and establishing semantic links between them [1]. Two interesting aspects that arise is the adequacy of the ontologies used for describing an entity in a dataset and the ease of identification of relations between description schemas. In the present paper, we describe a methodology for exploiting

[1] http://www.w3.org/2001/sw/
[2] http://www.w3.org/RDF/
[3] http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 117–122, 2012.

information on the instance level found in the World Wide Web in order to propose additional meaningful concepts for describing a class of objects in a dataset. Furthermore, the method allows the identification of similarities between different schemas and thus provides the basis for aligning their underlying ontologies. The method relies on producing association graphs between the entities present in the relations found from the information extraction process, identifying entities corresponding to an entity known from a LOD dataset and examining the presence of additional properties not present in the description schema used by the dataset.

The paper is structured as follows. First, we briefly present related work. Then, we provide a description of the Linked Data datasets that we used. In section 4, we present the process for acquiring web documents and retrieving relation tuples from them. In section 5, we describe the preliminary method for constructing a graph corresponding to the aforementioned relations and some observations on the results of this process. The final section reports our conclusions so far as well as the next steps for improving the tools involved and assessing the effectiveness of the proposed method.

2 Related Work

Ontology alignment is a key-technology for the purpose of interlinking different datasets, by discovering equivalent and/or semantically similar classes and properties between the ontologies of distinct repositories. Correspondences can then drive further associations between data in distinct repositories. Some examples of recent alignment systems are SAMBO [2], ASMOV [3] and RIMOM [4]. Given the wealth of instance-level data available in the LOD repositories, it is important to consider techniques based on acquiring knowledge from instances. In this regard, in relation to LOD, certain approaches aim to connect datasets at the conceptual level using LOD information as it is. The BLOOMS ontology alignment system [5] is such a system. It uses information existing in the Wikipedia hierarchy in order to bootstrap the alignment of two input ontologies. However, the BLOOMS approach is limited to the categorization of Wikipedia to perform the contextual match, not taking into account the possible association with other available classification systems.

Ontology enrichment relies heavily on processes like object identification, synonym resolution and relation extraction in order to expand an existing ontology with additional and/or specialized terms. Depending on the used approach, ontology learning systems can enrich a given ontology with both concepts and relations [6, 7], solely with concepts [8] or solely with relations [9, 10]. The purpose of our experiments is to exploit the structured nature of the data available in the LOD repositories in order to improve on both the relation and concept extraction from raw web content by associating the discovered information with the LOD.

3 The Jamendo and Magnatune Data Sets

Jamendo is a repository of music licensed under Creative Commons. Jamendo hosts 55451 albums; however, that dataset includes 5786 of them. For the purposes of our experiment, we used the RDF dump for the dataset, since the available web service seems unstable and became frequently unavailable. The dataset is interlinked with

GeoNames[4], a geographical database. The database is accessible via daily dumps and Web Services. For obtaining lexical information on the GeoNames entities included in Jamendo, we used the Java client for GeoNames Web Services.

Magnatune is an independent label, which follows the pay-as-you-wish business model for the creations of its artists. Part of the label's roster is described at the Magnatune dataset, in similar fashion with that of Jamendo. The dataset includes descriptions for 318 artists, 706 records and 17203 music tracks.

4 Information Extraction from Web Search Results

The first step in the process of acquiring a corpus of Open Web resources is to retrieve pages relevant to the space covered by the Jamendo dataset, or a subset of it. For the task, we used the Bing Search API 2.0. The main entity that we used for our search queries was the artist's name, retrieved from the `foaf:name` property within the RDF description of a given entity. When the name contained less than two words (excluding stop-words like "the", "a" etc.) we appended the query string with the term "music", in order to avoid large amounts of irrelevant results from the search engine.

The names were selected in the following manner: The artists whose descriptions included a `foaf:based_near` property, that is the artists linked with a GeoNames feature, were given priority. From the set of 3505 in the RDF dump, this restriction held for 3244 of them. These artists were selected as candidates for performing queries. For the construction of the query set, we indexed the remaining artist names and selected random indexes. For the Magnatune dataset, we randomly selected 200 artists without any further restrictions, since the amount of available descriptions was significantly lower. The search queries were then executed and we reserved the 50 first search results returned by the Bing Search API for further processing in both cases.

In order to get a text segment suitable for information extraction, we processed each page retrieved from the search session in the following manner: The raw HTML page was stripped from irrelevant information. This included formative content like HTML tags and scripting sections, as well as content that was irrelevant to the main text of the page, like menus, ads, lists of previous articles etc. We based the module for removing such elements on the boilerpipe library[5]. A step that was deemed necessary was the resolution of co-references within the text. The absence of such an analysis led to the production of numerous relations that were not useful since they associated entities that could not be resolved. Use of pronouns and generic terms, like "the band", "the group" do not allow the direct expansion of the relation set for an entity. To overcome this issue, we use the co-reference resolution module of the OpenNLP Tools[6]. Finally, we enforced the OpenNLP name entity resolution module in order to identify Named Entities within the obtained text segments. Lexemes recognized as named entities were given greater priority during the graph construction phase, as the probability of them being distinct object is greater.

---

4 http://www.geonames.org

5 http://code.google.com/p/boilerpipe

6 http://opennlp.sourceforge.net/projects.html

The documents obtained from the previous linguistic analysis were fed to the Open Information Extraction module. We used REVERB [11], a second generation OIE system. REVERB builds on the methods established in older systems, like TEXTRUNNER [12], by enforcing lexical and syntactic constrains in the extraction process. The triples that do not satisfy these constraints are not included in the returned relation set, thus the amount of irrelevant or erroneous relations is reduced. The corpora formed from the web search based on the two datasets were given as input to REVERB separately. Due to limitations of the boilerplate removal, some relations were not coherent, since they contained code snippets or HTML tags as arguments. We passed the initial results from a simple heuristic-based module in order to eliminate such relations. In the end, there were 506420 relations derived from the Jamendo search results and 438530 from the one for Magnatune.

5 Construction of the Object Graphs

Following the production of relations from the OIE module, the next step was to construct a graph denoting the associations between entities, as they occurred in the relation set. The central entities were the ones associated with the name of a musician, as retrieved from the datasets. Each of these entities constitutes a node in the graph. Going through the relation set, we retrieved every relation that involved the specific entity as either argument. These relations are the vertices from the given node to other entities. After repeating the step for every name, we constructed an unconnected graph that included every direct relation discovered for every musician.

The next step was to identify relations that are already in the RDF of the dataset. For each tuple (*E, rel, Arg*) or (*Arg, rel, E*), where *E* is the currently examined entity, *rel* is a lexicalization of an association and *Arg* is the other entity involved in the relation, we examined if *rel* is synonymous or similar to a property in the RDF. Specifically, we compared *rel* and the property names found within the LOD RDF using JWNL[7], a Java library for accessing WordNet[8]. If the two strings, after trivial manipulation like elimination of non-letter characters, are found to have a common sense in WordNet, the two relations are considered similar. If the argument *Arg* in the relation triple is also a lexicalization of the object linked with the entity *E* in the LOD dataset, we claim that the entity belongs to the dataset and the rest of the relations are possibly additional concepts for describing it.

An example of a sub-graph for a specific artist -with some indicative relations- is depicted in figure 1. The produced graph includes relations that exist in the RDF description from the Linked Data repository (e.g *(Beth Quist, recorded, Lucidity)* and meaningful relations that could be included in the description (is a vocalist and pianist, participated in the duo Ishwish). However, there are relations that have no concrete meaning (is an angel, includes perspective) and relations that are not sufficiently resolved (recorded her last solo album).

[7] http://sourceforge.net/projects/jwordnet/
[8] http:// wordnet.princeton.edu/

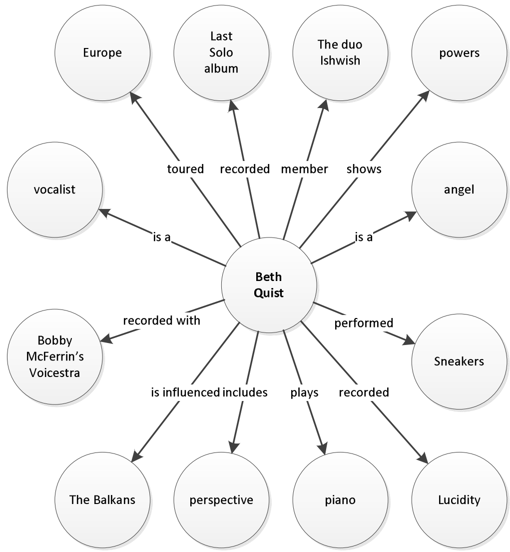


Fig. 1. Part of the association graph for the artist Beth Quist (http://dbtune.org/magnatune/artist/beth_quist)

The existence of a relation from or to an entity of known classification in multiple entities within the same dataset and between the two distinct datasets used for extracting web information is a good quantifier for increasing the probability that the property holds as a characteristic of the class and therefore it could be added in the ontology. For example, relations stating participation in a music group are frequent, so it is highly possible that membership is a meaningful property. On the contrary, a relation stating that musician A "is an angel" does not appear for multiple artists, therefore the probability of such a relation being meaningful is reduced.

6 Conclusions and Future Work

The purpose of our experiments at this stage was to examine the possibilities of acquiring knowledge in the conceptual level by combining web information and data available from LOD repositories in the instance level.

The preliminary results in a restricted domain (works of music) indicated that there is significant room for the introduction of additional properties not taken into account in the descriptions provided by the examined schemas. Furthermore, the equivalent properties between the two ontologies are revealed in a straightforward manner, though it is important to observe the inherent similarity of the schemas in the first place. To obtain more conclusive results, we will test the alignment opportunities in datasets with greater degree of differentiation.

For this run of the system, the hypothesis was that a lexicalization is considered as equivalent to a single entity. This held true in most of the cases, as the domain of the original search results was restricted from the beginning. However, there were examples where entity disambiguation should take place. For example, the music artist with the name “Cicadas” was associated with irrelevant results, as the insect with the same name produces “music” so the query did not differentiate the results for the band and the insect. For more generic terms the veracity of this statement was extremely low, as expected. The term “members” was a typical problematic word, since there were membership relations for multiple bands and these relations were included in the same sub-graph, connected with a single entity. In order to resolve this issue, we will expand the OIE system used in order to keep track of the relations derived from each web page and associate occurrences of relations including the term with a specific artist, the one for which the page is relevant. After the improvements in the infrastructure of our system and the distinct processes involved, we will apply a strict, formal graph similarity method in order to quantify the distance between the graphs derived from the relation triples and the LOD description graph.

References

1. Bizer, C., Heath, T., Berners-Lee, T.: Linked data – the story so far. Intl J. on Semantic Web and Information Systems 5(3), 1–22 (2009)
2. Lambrix, P., Tan, H.: SAMBO – a system for aligning and merging biomedical ontologies. Journal of Web Semantics 49(1), 196–206 (2006)
3. Jean-Mary, Y.R., Shironoshita, E.P., Kabuka, M.R.: Ontology matching with semantic verification. Journal of Web Semantics 7(3), 235–251 (2009)
4. Li, J., Tang, J., Li, Y., Luo, Q.: Rimom: A dynamic multistrategy ontology alignment framework. IEEE Transactions on Knowledge and Data Engineering 21(8), 1218–1232 (2009)
5. 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)
6. Alani, H., Sanghee, K., Millard, E.D., Weal, J.M., Lewis, P.H., Hall, W., Shadbolt, N.: Automatic Extraction of Knowledge from Web Documents. In: Proc. of HLT (2003)
7. Kim, S.-S., Son, J.-W., Park, S.-B., Park, S.-Y., Lee, C., Wang, J.-H., Jang, M.-G., Park, H.-G.: OPTIMA: An Ontology Population System. In: 3rd Workshop on Ontology Learning and Population (2008)
8. Etzioni, O., Kok, S., Soderland, S., Cagarella, M., Popescu, A.M., Weld, D.S., Downey, D., Shaker, T., Yates, A.: Unsupervised named-entity extraction from the Web: An experimental Study. Artificial Intelligence 165, 91–134 (2005)
9. Brewster, C., Ciravegna, F., Wilks, Y.: User-Centred Ontology Learning for Knowledge Management. In: Andersson, B., Bergholtz, M., Johannesson, P. (eds.) NLDB 2002. LNCS, vol. 2553, pp. 203–207. Springer, Heidelberg (2002)
10. Suchanek, F.M., Ifrim, G., Weikum, G.: LEILA: Learning to Extract Information by Linguistic Analysis. In: Proc. of the 2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge, Sydney, Australia, pp. 18–25 (2006)
11. Etzioni, O., Fader, A., Christensen, J., Soderland, S.: Mausam: Open Information Extraction: the Second Generation. In: Intl. Joint Conference on Artificial Intelligence (2011)
12. Yates, A., Cafarella, M., Banko, M., Etzioni, O., Broadhead, M., Soderland, S.: TextRunner: Open Information Extraction on the Web. Computational Linguistics 42, 25–26 (2007)

Modeling the Context of Scientific Information: Mapping VIVO and CERIF

Leonardo Lezcano[1], Brigitte Jörg[2], and Miguel-Angel Sicilia[1]

[1] Information Engineering Research Unit, Computer Science Department, University of Alcalá, Polytechnic Building, Ctra. De Barcelona km. 33.6 28871 Alcalá de Henares (Madrid), Spain
{leonardo.lezcano,msicilia}@uah.es
[2] DFKI GmbH, Language Technology Lab Projektbüro Berlin, Alt-Moabit 91c 10559 Berlin, Germany
brigitte.joerg@dfki.de

Abstract. Institutional repositories (IR) and Current Research Information Systems (CRIS) among other kinds of systems store and manage information on the context in which research activity takes place. Several models, standards and ontologies have been proposed to date as a solution to give coherent semantics to research information. These present a large degree of overlap but also present very different approaches to modeling. This paper presents a contrast of two of the more widespread models, the VIVO ontology and the CERIF standards, and provides directions for mapping them in a way that enables clients to integrate data coming from heterogeneous sources. The majority of mapping problems have risen from the representation of VIVO sub-hierarchies in CERIF as well as from the representation of CERIF attributes in VIVO.

Keywords: CERIF, VIVO, CRIS, research information, scientific information, ontologies, knowledge representation, mapping.

1 Introduction

Traditionally, most of research has been curiosity-led, discipline-oriented, and motivated and executed by a small group of individuals following hypothesis, experiment or proven method. Nevertheless, the complex problems that science is facing nowadays require large teams with each member having a specialized contribution to the whole. These collaborative teams are often geographically dispersed and belong to different disciplines. Gibbons et al. refer to the fact that science has been shifting from discipline-oriented to cross-disciplinary research as Mode Two [1].

Increased knowledge, the paradigm shift, recognition of economic stimulus and collaborative interdisciplinary science lead inexorably to the need for systems to assist researchers, administrators, strategists, opinion-formers, entrepreneurs and also the general public [2]. Current Research Information Systems (CRIS) are expected to provide such scientific information. In order to support decision-making and

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 123–129, 2012.

knowledge-creation, CRIS can be used to find specialized equipment or facilities, recognize innovations and results (to avoid duplication of effort), manage the grant process, produce statistics and reports, evaluate projects and assess science, promote science in society and to locate funding sources, among other applications. In order for research information systems to properly represent the content and context of research work, Sicilia [3] provides examples that could serve as a point of departure to develop an upper ontology for research methods and tools.

CERIF [4] is the common european research information model for the development of new CRISs and a template both for data exchange between CRISs and for mediating access to multiple heterogeneous distributed CRISs. CERIF has been released as an EC Recommendation to European Member States in 2000.

On the other hand, the VIVO project [5] is creating an open, Semantic Web-based network of institutional ontology-driven databases to enable national discovery, networking, and collaboration via information sharing about researchers and their activities.

The purpose of the present research is to study the overlaps and differences between these two widespread approaches to research information modeling. Section 2 provides a background of both models. Then, Section 3 explains the directions for mapping them in a way that enables clients to integrate data coming from heterogeneous sources. Conclusions are finally presented in Section 4.

2 Background

This Section introduces the VIVO ontology and the CERIF model.

2.1 The CERIF Model

CERIF is considered a standard recommended by the European Union to its Member States[1]. The CERIF model represents information about entities such as Publication, Project, Organization, Person, Product, Patent, Service, Equipment and Facility as well as semantically enhanced relationships between these entities in a formalized way. The physical model is a relational database model available as SQL scripts based on common ERM (Entity Relationship Model) constructs. The latest releases include a formalized, so called “Semantic Layer,” and an XML interchange format [6]. The CERIF model is conceptually structured into entity types and features. Among the types, *core*, *result*, *link*, and *2nd level entities* are distinguished. *Multilingualism* and *semantics* are considered as features. Further details can be found in [7]. A mapping between the CERIF part related to published results of scientific research and the MARC 21 bibliographic standard is studied in [8], and a CERIF data model extension for the evaluation of scientific research results is proposed by Ivanovic et al. in [9].

2.2 The VIVO Ontology

All data in VIVO are represented as RDF statements using classes and properties from OWL ontologies. These ontologies specify the types of resources described in VIVO and

[1] http://cordis.europa.eu/cerif/

their relationships. The VIVO core ontology[2] models the people, organizations, and activities involved in scientific research. According to the linked data initiative, the VIVO core ontology extends existing ontologies such as the Friend-of-a-Friend (FOAF) ontology[3], which provides the basis for describing persons and organizations, and the Bibliographic Ontology (BIBO)[4]. A comprehensive list of the ontologies integrated into the VIVO Ontology can be found in VIVO Project Wiki[5]. A description of VIVO ontology design principles like remaining independent of specific domains and representing temporal relationships is also available in the VIVO Project Wiki.

3 Mapping CERIF and VIVO

This document is intended to provide mapping recommendations for the elements of the CERIF model described in the FDM specification document [7] to the VIVO 1.4 ontology. Both the entire CERIF model and VIVO allow for many more types of relationships and entities, however, it is expected that the approach required to create any mapping between such knowledge artifacts can be derived from the following recommendations. General metrics from CERIF and VIVO are provided in Table 1 and 2, and discussed below in the Conclusions section.

Table 1. CERIF model metrics

CERIF			
Entities	**Attributes**	**Link Entities**	**Language Entities**
56	1766	120	61

Table 2. VIVO ontology metrics

VIVO		
Classes	**Datatype Properties**	**Object Properties**
209	94	218

3.1 CERIF Base, Result and Infrastructure Entities ([base], [result] & [infra])

Mapping CERIF Base, Result and Infrastructure entities to VIVO is a straightforward process given the fact they have no foreign key (FK)[6] and therefore most of their attributes can be mapped as datatype properties between a given class in VIVO and a data literal and the rest. It should be noted that a minority of the attributes, like cfURI in the cfProj table, are mapped to an object property in VIVO like webpage, but the pointed object, in this case an instance of URLLink, plays a role of user-defined datatype rather than an instance of a complex object. At the same time, CERIF uses "2nd level entities" to represent complex objects, which are connected to base entities through link entities. These have been mapped to VIVO as shown in Section 1.3. From a conceptual perspective, "2nd level entities" in CERIF can be considered as the environment in which the base entities act and communicate, and produce results. The cfEvent table is currently classified by CERIF as a 2nd level entity while VIVO considers the Event class as an independent piece of information.

[2] http://vivoweb.org/download
[3] http://www.foaf-project.org/
[4] http://bibliontology.com/
[5] http://sourceforge.net/apps/mediawiki/vivo/index.php?title=Ontologies
[6] The Currency Code attribute (*cfCurrCode*) is an exception.

Table 3. Examples of mappings between CERIF Base and Result Entities and VIVO classes and properties

CERIF		VIVO	
Table	**Attribute**	**Property**	**Class**
cfProj	cfURI	webpage only URLLink	Project
	cfAcro	description only Literal	
	cfStartDate	dateTimeInterval only DateTimeInterval	
	cfEndDate		
cfOrgUnit	cfAcro	abbreviation only Literal	Organization
	cfURI	webpage only URLLink	
	cfHeadcount[7]	not modeled (Assuming that all members are in the system, it can be inferred by counting the number of Person instances which are related to a given Organization through the hasCurrentMember property)	
	cfTurn	not modeled (not even in PrivateCompany)	
cfPers	←-------------------------------→		Person
cfResPubl	←-------------------------------→		Document
cfResPat	←-------------------------------→		Patent
cfResProd	←-------------------------------→		CaseStudy, Dataset, etc.
cfFacil	←-------------------------------→		Facility
cfEquip	←-------------------------------→		Equipment
cfSrv	←-------------------------------→		Service

Result entities like cfResPubl, cfResPat and cfResProd can be mapped to specializations of the InformationResource VIVO class. Table 3 includes mapping examples. As in the first two cases, once the mapped classes and tables have been identified, the mapping granularity must be further increased by mapping CERIF attributes to VIVO properties. It should be noted that most of the Multiple Language CERIF features are not modeled in VIVO, for example, although VIVO allows for several abstracts, keywords and titles to be assigned to the same article, the language of such texts has not been defined.

3.2 CERIF Semantics [class]

In the CERIF model, the semantics of a given record within a broad entity like Project (cfProj_Class) are enriched by a time-stamped reference to the CERIF Semantic Layer to host any vocabulary, e.g. the CERIF 1.3 Vocabulary[8]. The VIVO ontology uses a sub-hierarchy to accomplish such specialization of concepts, e.g., *Human Study* is a subclass of *Research Project* which in turn is a subclass of the top classification *Project*. More examples are included in Table 4.

---

[7] A contact with the CERIF task group indicated that, in the next major CERIF release, measurement attributes such as headcount and turnover will not be supported anymore explicitly. The recommendation will rather be towards using the new and generic measurement entities for calculations, and other inferred data.

[8] http://www.eurocris.org/Uploads/Web%20pages/CERIF-1.3/Semantics/CERIF1.3_Vocabulary.xls

Table 4. Examples of mappings between CERIF Semantics and VIVO classes and subclasses

CERIF		VIVO	
Table	**Class Term**	**SubClass**	**Top Class**
cfProj_Class	Discipline Codes, Application Codes	Research Project, Human Study or Clinical Trial from **eagle-i**.	Project
cfPers_Class	Consultant, Lecturer, Research Fellow, etc	Faculty Member, Librarian, Non-Academic Staff, etc.	Person (**FOAF**)
cfOrgUnit_Class	Private non-profit, University College, etc.	Association, College, Consortium, etc.	Organization (**FOAF**)
cfResPubl_Class	Book, Review, etc.	BIBO and VIVO classes	Information Resource

3.3 CERIF Link Entities [link]

CERIF defines every relationship between two entities using a pair of records identifiers (cfId1 and cfId2) taken from the tables representing those entities. The semantics of the pair are then enriched by a time-stamped reference to the CERIF Semantic Layer to host vocabularies of any structure. The VIVO ontology fulfills this relation classification task by means of a hierarchy of *Object Properties* combined with the taxonomy of VIVO classes. The mapping of the cfFraction attribute (Float) involves particular mechanisms depending on the semantics of each relation. For example, the *authorRank* property allows representing the cfFraction value related to the author-publication link, while the *description* property can be used to map the values of the cfFraction attribute in the person-project link. In both cases there are VIVO classes (i.e., Authorship and ResearcherRole) which are oriented to describe such relations and serve as domain classes for the mentioned properties (see Table 5).

Table 5. Examples of mappings between CERIF Link Entities and VIVO classes and properties

<table>
<tr><th colspan="4">CERIF</th><th colspan="3">VIVO</th></tr>
<tr><th>Table</th><th>cfId1</th><th>Semantic Stamp</th><th>cfId2</th><th>Domain</th><th>Object Property</th><th>Range</th></tr>
<tr><td>cfProj_ResPubl</td><td>cfProjId</td><td>Originator</td><td>cfResPublId</td><td>Project</td><td>Information Product</td><td>Information Resource</td></tr>
<tr><td>cfProj_Fund</td><td>cfProjId</td><td>Funder</td><td>cfFundId</td><td>Project</td><td>hasFunding Vehicle</td><td>Agreement</td></tr>
<tr><td>cfProj_OrgUnit</td><td>cfProjId</td><td>Coordinator</td><td>cfOrgUnitId</td><td>Project</td><td>realizedRole</td><td>Role</td></tr>
<tr><td>cfProj_OrgUnit</td><td>cfProjId</td><td>Fract[0.2]</td><td>cfOrgUnitId</td><td>Project</td><td colspan="2">not modeled</td></tr>
<tr><td rowspan="3">cfPersName_Pers</td><td rowspan="3">cfPersId</td><td rowspan="3">Spelling Variant</td><td rowspan="3">cfPersId2</td><td rowspan="3">Person</td><td>firstName (dataype)</td><td rowspan="3">not defined</td></tr>
<tr><td>middleName (datatype)</td></tr>
<tr><td>lastName (dataype)</td></tr>
<tr><td rowspan="2">cfPers_ResPubl</td><td rowspan="2">cfPersId</td><td rowspan="2">Author</td><td rowspan="2">cfResPublId</td><td>Person</td><td>authorIn Authorship</td><td>Authorship</td></tr>
<tr><td>Authorship</td><td>Information Resource InAuthorship</td><td>Information Resource</td></tr>
</table>

Table 5. *(Continued)*

cfPers_ResPubl	cfPersId	Author (percentage)	cfResPublId	Authorship	authorRank (datatype)	integer
cfPers_OrgUnit	cfPersId	Affiliation	cfOrgUnitId	Person	current MemberOf	Organization
cfPers_OrgUnit	cfPersId	Sub Affiliation	cfOrgUnitId	Person	current MemberOf	Organization
				Organization	hasSub Organization	Organization
cfPers_OrgUnit	cfPersId	Board-Member or TG-Leader	cfOrgUnitId	Person	hasLeaderRole	LeaderRole
				LeaderRole	roleContributesTo	Organization
				Person	currentlyHeadOf	Organization
cfProj_Pers	cfPersId	Coordinator	cfProjId	Person	hasOrganizerRole	OrganizerRole
				OrganizerRole	roleRealizedIn	Project
cfProj_Pers	cfPersId	Coordinator [fract=0.7]	cfProjId	OrganizerRole	description	Literal
cfProj_Pers	cfPersId	Participant	cfProjId	Person	hasResearcherRole	ResearcherRole
				ResearcherRole	roleRealizedIn	Project
cfProj_Pers	cfPersId	Participant [fract=0.3]	cfProjId	ResearcherRole	description	Literal
cfOrgUnit _PAddr	cfOrg UnitId	post-office-box	cfPAddrId	Organization	mailingAddress	Address
cfOrgUnit _EAddr	cfOrg UnitId	Email	cfEAddrId	Organization	email	not defined
cfOrgUnit _EAddr	cfOrg UnitId	Skype	cfEAddrId	not modeled		

4 Conclusions

Information models and knowledge artifacts have been designed and improved in the last decade to represent the research domain. In particular, CERIF and VIVO have been widely adopted for such purpose. As an approach to support interoperability and integration of the systems based in these models, this paper makes a comparison study between the CERIF relational database model and the VIVO ontology. A challenging task during the study has been that of properly mapping the information semantics represented in the CERIF Semantic Layer to the VIVO semantics supported by OWL.

Broadly analyzing the interoperability of both models reveals that VIVO does not support *multilingual* features as CERIF does. On the other hand, without considering the CERIF entities that are exclusively oriented to support the language features, the 209 VIVO classes provide a higher classification granularity than the 56 CERIF entities. Similar conclusions are reached when comparing the 218 Object Properties and sub-properties in the VIVO ontology with the 120 link entities in CERIF (see Table 1 and 2). While offering more classes and relationships improves the semantics and accuracy of the research knowledge representation, it should be noted that maintainability and integration feasibility may be jeopardized. In order to increase the semantics associated to entities while preserving the simplicity of the model, CERIF use a controlled vocabulary to describe entities and relationships (see Section 3.2). Nevertheless, it is a flat classification method that does not support attributes inheritance.

The detailed mapping recommendations in Section 3 show that the most significant research information can be successfully converted from one representation to the other and vice versa. In fact, the three main entities in the CERIF model (i.e., Person, Project and OrganizationUnit) and their attributes can be straightforwardly mapped to three classes that also play an essential role in the VIVO ontology (i.e., Person, Project and Organization). At the same time, the study has found some particular cases where modeling at one side does not support a given piece of information from the other side. Having 1700 attributes allows the CERIF model to represent very specific information like for example the Skype user assigned to a Person.

Acknowledgments. The work leading to these results has received funding from the European Commission under grant agreement nº 250525 corresponding to project VOA3R (Virtual Open Access Agriculture & Aquaculture Repository: Sharing Scientific and Scholarly Research related to Agriculture, Food, and Environment), http://voa3r.eu

References

1. Gibbons, M.: The new production of knowledge: the dynamics of science and research in contemporary societies. SAGE (1994)
2. Zimmerman, E.: CRIS-Cross: current research information systems at a cross-roads. In: Proceedings: Gaining Insight from Research Information, at the 6th International Conference on Current Research Information Systems, Kassel, Germany (2002)
3. Sicilia, M.-Á.: On Modeling Research Work for Describing and Filtering Scientific Information. In: Sánchez-Alonso, S., Athanasiadis, I.N. (eds.) MTSR 2010. CCIS, vol. 108, pp. 247–254. Springer, Heidelberg (2010)
4. Jörg, B.: CERIF: The common European research information format model. Data Science Journal 9, 24–31 (2010)
5. Krafft, D.B., Cappadona, N.A., Caruso, B., Corson-Rikert, J., Devare, M., Lowe, B.J.: Vivo: Enabling national networking of scientists. In: Proceedings of the WebSci 2010: Extending the Frontiers of Society On-Line, Raleigh, US (2010)
6. Jörg, B., Ojars, K., Jeffery, K., van Grootel, G.: CERIF XML 2008 – 1.0 Data Exchange Format (2009)
7. Jörg, B., Jeffery, K., Dvořák, J., Houssos, N., Asserson, A., van Grootel, G.: CERIF 1.3 Full Data Model (FDM) - Introduction and Specification (2012)
8. Ivanovic, D., Surla, D., Konjovic, Z.: CERIF compatible data model based on format. The Electronic Library 29, 52–70 (2011)
9. Ivanovic, D., Surla, D., Racković, M.: A CERIF data model extension for evaluation and quantitative expression of scientific research results. Scientometrics 86, 155–172 (2010)

An Attempt of the Heuristic Evaluation of Visualization in Searching Economic Information in Topic Maps

Helena Dudycz

Wroclaw University of Economics, Department of Information Technologies,
Komandorska Str. 118/120, 53-345 Wroclaw, Poland
helena.dudycz@ue.wroc.pl

Abstract. The usefulness of economic indicators in assessing functioning of an enterprise depends on comprehension by decision-makers of semantic connections existing between ratios. Analysis of economic indicators with regard to semantic relations often has essential impact on formulating accurate conclusions. However, the knowledge of semantic connections and resulting from them information concerning functioning of an enterprise is often possessed only by experienced financial analysts. Topic map standard allows creating a model of knowledge representation, searching and acquiring unique information. The creation of topic map for the ontology of economic indicators facilitates inter alia analyzing economic ratios on account of semantic connections between them. In the semantic search in topic map the visualization plays important role. We have audited the heuristic evaluation of the visualization in searching information on economic indicators based application for Return on Investment indicator. The research with participation of users was carried out.

Keywords: ontology of economic indicators, information visualization, evaluation of usability of visualizing in searching information, topic map, heuristic evaluation.

1 Introduction

The essence of examining and assessing functioning of an enterprise consists in appropriate calculating and using economic indicators coming from various financial reports. Their usefulness depends also on comprehension by decision-makers of hierarchic and semantic connections existing between indicators. Analysis of semantic relations often has essential impact on formulation of accurate conclusions from the economic analysis assessing the functioning of an enterprise. However, the knowledge of semantic connections and resulting from them information concerning functioning of an enterprise is often possessed only by experienced financial analysts. Research carried out by E. Awad i H. Ghaziri, published in 2004, confirms formulated in 1995 by I. Nonaka and H. Takeuchi statement that as much as 95% of information is preserved as tacit knowledge [14, p. 209]. The research on connecting two essential issues is conducted [4]:

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 130–142, 2012.

- knowledge representation in information system, that would also enable to change tacit knowledge into explicit knowledge,
- using, besides traditional information, searching methods solutions allowing contextual searching based on the visualization's methods.

More and more attention is paid on using semantic technologies such as topic map (TM) standard [ISO/IEC 13250:2003]. Topic Map is "model of knowledge representation, and its main purpose is to organize the information through semantic linkage in the data, concepts, and sources" [13, p. 30]. It can be used to the representation of complex structures of knowledge bases [2], the delivery of a useful model of knowledge representation [12 , p. 174]; [13, p. 30]), search and acquire unique information [20]. TM is a relatively new form of the presentation of knowledge, which put emphasis on data semantics and ease of finding desired information (see also [1]; [13, p. 30]), where multiple contextual indexing can be used. TM can represent information using topics (representing any concept), associations (which represent the relationships between them), and occurrences (relationships between topics and information resources relevant to them) (e.g. [12, p. 175). TM can support the extraction of information from all relevant subsystems existing in the enterprise, because allows for integrating all sorts of data sources [1]. The ability of TM to link resources anywhere and to organise these resources according to a single ontology will be a key component of knowledge management solutions [12, p. 174].

We are conducting the research on the usage of the topic map as a model of knowledge concerning the analysis of economic ratios (see e.g. [4]). TM standard allows showing taxonomic relations and semantic connections both binary and n-nary. Creating topic maps, which include relations between different economic indicators, allows inter alia an analysis of the value synthetically indicator having knowledge about connected by semantic relationships of the values of another indicators. The conception of the usage of the topic map as a model of knowledge of the analysis economic indicators has the advantage that created model of ontology can be relatively easily modified and simultaneously it is possible to use multiple applications based on different ontologies of a particular area. This is essential, because there is no single universal system of economic ratios, which would be used in all enterprises.

We are carrying out research the aim of which is the visual exploration of data from all relevant subsystems existing in the enterprise by decision-makers using TM as a model of economic knowledge representation. The main goal of this paper is to discuss the research on using visualization of semantic network in searching economic information. In this study we used the application of an ontology for return on investment (ROI) indicator. The article is structured as follows. The next section presents shortly the related work. In Section 3 we briefly describe the design of the research. In Section 4, the analysis of results of the research and conclusions are presented. Finally, in the last section a summary of this work is given and future research projects are indicated.

2 Visualization in Searching Information in Topic Map

Topic map is a semantic graph that contains definitions of a set of topics and a set of association between topics. Visualization in TM allows navigation from topic to topic in a highly interactive manner: interesting nodes can be put in the foreground with zooms, translations and rotations. Users can delete inapplicable branches of the tree or expand interesting ones. Fundamental factors for a good visualization of the application of TM are: the overview of the structure for the global understanding of the structure and of the relationships within the hierarchy; the ability to zoom and to select some nodes; and dynamic requests in order to filter data in real time [7]. TM can be used to represent economic knowledge about indicators, where visualization can assure semantic information search and facilitate interpretation data by non-technically-minded users. Thanks to the visualization users can more swiftly notice and understand various structural and semantic relations.

B. Shneiderman defined that an effective visual exploration tool should allow to do the following tasks in an easy-to-use manner: overview, zoom, filter, details-on-demand, relate, extract, history (e.g. described in [18, pp. 165-166]). He indicated these tasks basing on observations of the behaviour of analysts, who attempt to extract knowledge from the data [18, p. 165]. TM software with a visualization module allows to do all tasks defined by B. Shneiderman except history. The functionality of TM application enables also to realize three paradigms of information retrieval visualization formulated by J. Zhang (thoroughly described in [21, p. 16]): (1) the Query searching and Browsing paradigm, (2) the Browsing and Query searching paradigm and (3) the Browsing Only paradigm.

One of the most pressing questions about visualization-based information retrieval systems is: “Can people use them?” (see [10, p. 824]). This question can also be set to the semantic searching in topic maps for the analysis of economic ratios, where the visualization plays important role. Systems that enable information retrieval should be intuitive to use or easy to interpret by the users of the system. A good interface of the information visualization contains a good representation (helps users identify interesting sources) and efficient navigation (allows users to access information quickly) [8]. The basic assumption of the navigation is that users should be able to view focus and context areas at the same time to present an overview of the whole knowledge structure. It is very important, because that how decision makers perceive and interact with a visual representation, can strongly influence their understanding of the data as well as the usefulness of the visual presentation [9, p. 48].

The visual analysis of economic indicators due to the semantic relationships can bring measurable benefits because semantic search is more efficient than that based on basic hierarchic structure (see [6]; [22, p. 1899]). Furthermore, the latest research points out also that searching information basing on semantic connections in topic map has a positive influence on discovering the essential information (see [19, p. 301]). Thanks to that decision-makers will receive a tool the use of which can result in easier acquisition of needed information from that existing in enterprise databases.

TM application may be a useful tool in the visual searching of information for decision-makers. However, it requires carrying on a research. So far we have studied: TM as the useful model of economic knowledge representation, and TM as the useful visualization in searching and acquiring unique information.

3 The Design of Research

The goal of our research is the visual exploration of information using the semantic searching based on the applications of TM, which contain the models of knowledge of the analysis of economic indicators. This study has been carried out in the two main phases and the five stages (Fig. 1).

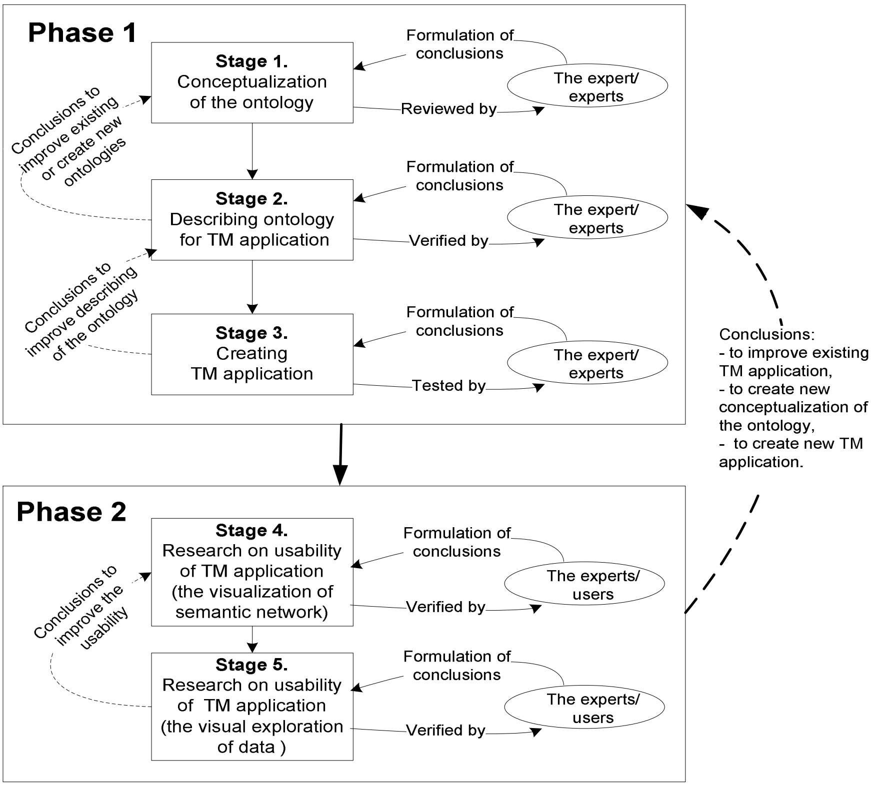


Fig. 1. The scheme of the design of research on the visual exploration of information using the semantic searching based on the applications of TM

Phase 1: Conceptualization of the ontology of economic indicators and creating TM application.

The goal of this study is verifying the concept of using TM standard for the model of knowledge about analysis economic indicators. This phase consists of the three stages. Results of this work are the following inter alia: the creation of the ontology for the selected analysis economic indicators; the creation of TM application for specific field of analysis of economic indicators; the creation of the procedure of building TM applications for the ontology of analysis economic indicators (based on the analysis of existing methods and methodologies); and the verification of the visualization semantic network in TM for searching and acquiring unique information on economic indicators.

Phase 2: Research on the usability of the visualization in economic information retrieval in TM application.

We have two aims of this study. The first goal is verifying the usability of applying the visualization semantic network in TM in searching and acquiring unique information in the analysis of economic indicators. The second goal is verifying the visual exploration of data from all relevant subsystems existing in an enterprise by decision-makers using TM as a model of economic knowledge representation. This phase consists of the two stages. Research in this phase is carried out on the basis of the model proposed by E. Brangier (the usage – adaptation – re-engineering cycle), "which highlights how human adaptations (of the users) are a source of innovation to design new uses" [5]. These studies enable to identify users' needs precisely and may contribute to the development of innovations. Results of this work are the following inter alia: the assessment of the usability of visualizing a semantic network based on TM; the solving of the problem of large number of topics in using TM as a tool enabling a visual exploration of a huge, complex and multidimensional data set.

In this article we focus on the presentation of research conducted in the second phase (stage 4).

4 The Heuristic Evaluation of Visualization in Searching Economic Information – The Research Description and the Analysis of Result

TM can be one of various tools for information visualization, which – as visual interactive interface – allows to search of information for decision-makers. However, it requires carrying on a research of usability of visualization in searching information. The standard ISO 9241-11 defines usability of IT systems as the three components: effectiveness, efficiency and satisfaction. A well-designed user interface inter alia facilitates the performance of user's task (effectiveness) and provides the pleasure of use (satisfaction). B. Shneiderman defined five measures essential in studying the system usability: system familiarity time, task performance speed, errors in task assignments, system feature retention, and subjective satisfaction [15]. In the research on evaluation of visualization in searching economic information following parameters are important: information retrieval speed, errors in interpretation of data (errors in task assignments) and subjective satisfaction.

In literature many methods of research and evaluation of the system usability and human-computer interaction are described (see inter alia [11]; [16]; [17]). The study of a prototype is conducted with the experts' participation (e.g. heuristic evaluation of user interface) and/or users (e.g. task-based user test, eye tracking).

4.1 Assumptions of the Research

We have decided to carry out the study with participants and using the heuristic evaluation and task-based user test. We have planned that none of the participants had searched information basing on the visualization of ontology before and the training has been restricted to minimum. Our research was also to tell how easy and clear is searching information with the use of the visualization of the semantic network, for a user who is not familiar with the topic map application.

For the comparison and verification of the results of this study we have decided to conduct two tests for two different TM applications. The first application contained the ontology of faculty “Business Informatics” at Wrocław University of Economics. The topics and associations existing between them in this ontology were known to all the participants. The second application contained the ontology of ROI indicators in Du Pont model. This area was known only to some of the participants.

In the research we used different TM applications, so we worked out two questionnaires which contained three parts. They differed from each other only in the first fragment, in which tasks to be done by the user, consisting in searching information, are listed.

Before starting the research on system utility we created an ontology for ROI indicator, built TM application in TM4L software and tested visual searching information. Setting about research with participants we decided to create application in Protégé 4.1 beta for ROI indicator, because there were difficulties with correct executing TM4L software in operation system MS Windows 7. In Protégé 4.1 beta, there is the module OntoGraf, which turned out to be sufficient to carry out the initial research, the aim of which was to verify the usefulness of visual semantic network in searching economical information, that is contextually connected.

4.2 Participants of the Research

It was decided to carry out a research with the participation of users. Their selection was not random, as they were to fulfil a double role. First role was to be a typical user, performing specific tasks in a topic map application for ROI indicator (research using the usability testing technique). Second role was to be an expert evaluating the usability of applied interface (research using heuristic evaluation of user interface).

The selection of the participants of the research allowed obtaining a group of people, who had various experience and knowledge concerning economy and analysis of economic ratios as well as systems and information technology. In the study 42 people participated: with only computer education (48% of all participants), computer science and econometrics education (21% of all participants) and non-computer education (31% of all participants, but in this group 80% of participants have economic education). None of them either searched information basing on the visualization of ontology before or was familiar with the program Protégé.

4.3 The Course of the Research

Performing tasks by the users was preceded with an introduction (15 to 20 minutes, depending on the group), in which it was shown: how to open the application with an ontology in the program Protégé, how to save the chosen fragment of the graph as a graphic file or as a graph, and what is the idea of semantic search and topic map standard. Afterwards the participants began the study. Firstly participants of the research accomplished seven tasks, consisting in searching information in the semantic network built for the faculty “Business Informatics”. Then the participants carried out seven tasks in semantic network for ROI indicator. For both applications they performed expert opinion of the usability of the applied interface according to identical criteria. We planned out 60 minutes on performing all assignments by the participants (that is 14 tasks and filling in two short questionnaires concerning heuristic assessment of the system). Many participants finished performing tasks and filling in questionnaires in shorter time (by about 10 minutes). Only 15% of all persons needed 70-75 minutes to accomplish their assignments. The research was not recorded by video camera, but during performing tasks participants were observed and their remarks were noted.

4.4 Analysis of the Results

In the research we used two questionnaires (for two applications) which contained three parts. The results of the study may be divided into two groups. First group results from the research using the usability testing technique. It concerns the correctness of performing tasks by users and the assessment of easiness of searching information basing on the visualization of the semantic network. These are the data obtained from the first part of the questionnaire. Second part of the results comes from the research using the heuristic evaluation of user interface. These are the data obtained from the second and third parts of the questionnaire. Results obtained from the first part of the questionnaire we thoroughly described in [3]. In this article we concentrate on presenting and elaborating the results obtained from using the heuristic evaluation of the usability of the visualization in searching economic information. The questionnaire contained:

- four criterions of an interface evaluation, in which users chose one of the following answers: *highly satisfactory*, *satisfactory*, *average*, *unsatisfactory*, *very unsatisfactory*;
- seven potential problems, for which participants of research chose one of the following answers: *no problem*, *a small problem*, *an important problem.*

In Appendix Table 1 there are data concerning the evaluation of the interface of searching information for the ontology of ROI indicator. In the context of the research particularly essential criteria are: an evaluation of the system in terms of visual clarity (criterion A) and an evaluation of the way of searching information based on the visualization of a semantic network (criterion D).

In the case of criterion A, 46% users assessed the proposed solution as *highly satisfactory* or *satisfactory*, whereas in the case of criterion D – 43% participants. In criterion D no one chose the option *very unsatisfactory*. For criterion A only 2% of participants stated that the proposed solution is *very bad*. Worst of all was estimated the usefulness of the interface on account of criterion B, as 5% chose answer *very unsatisfactory*, but as many as 40% – *unsatisfactory*. This result is influenced also by the evaluation of the interface of the Protégé software.

During the analysis of data in Appendix Table 1, we can notice that users with computer education evaluated the applied interface for searching information most critically, whereas users with non-computer education most positively. This group assessed best the interface on account of criterion C (62%), whereas two other groups on account of criterion A (with computer education – 40%, whereas with computer science and econometrics education – as many as 78%).

In Appendix Table 2 there are data concerning the identification of potential problems connected with the usage of the visualization of the semantic network in searching economic information. In the context of conducted research particularly essential are results concerning problems no. 4, 5 and 6. In literature the possibility of occurring difficulties in finding information basing on visualization of semantic network due to a big number of points, lines connecting them and use of many colors is indicated. In research carried out for the problem no. 6 (*too many colors on the screen*) as many as 60% of participants answered that *no problem*. For the problem no. 5 (*the difficulty in reading information on screen*) 45% of participants chose answer: *no problem*. However, for the problem no. 4 (*find necessary information*) and problem no. 3 (*understanding the relation between information on the screen and the executed operation*) only 26% of participants chose that answer. It is the least value amongst all listed problems in the questionnaire on account of assessment: *no problem*. the analysis of data obtained for problems no. 3 and 4 in comparison with the results from the other parts of questionnaires results in the following conclusion: in the next research longer time for training in the idea of searching information basing on visualization of semantic network will be needed.

However, analyzing seven distinguished potential problems in terms of assessment: *an important problem*, quite promising results were obtained: from 7% (problem no. 2) to 17% (problem no. 3). For problems especially interesting us, that is no. 4, 5 and 6, it comes to 14, 12 and 16% Respectively. Despite quite optimistic results, the data obtained for problems no. 3 and 4 prove the necessity of conducting training (about 30 minutes) before the next research with participants.

We will now analyze this data in terms of education of research participants. Among group of people with computer education, the highest percentage of the answer *an important problem* was received for the problem no. 6 (25%), whereas the highest percentage of the answer *noproblem* for problems no. 5 and 7 (55%). In the group with computer science and econometrics education most participants chose: *an important problem* for problems no. 3 and no. 5 (33%). Meanwhile no one from this group marked for the problem no.7: *an important problem*. In the opinion of the participants from this group the least problem is 4 (45%). In the last analyzed group of the participants, that is with non-computer education most users chose the answer *an important problem* for the problem no. 4 (15%). Meanwhile no one stated that the problem no. 5 *is an important problem*. According to this group the least problem of all is the problem no. 6 (as many as 84%).

4.5 Conclusions and Next Steps

The results obtained from the research described herein are promising. They confirm initial conclusions obtained from the analysis of task-based user test (described in [3]). This study indicates that searching information basing on the visualization of the semantic network can be a useful tool for decision-makers. Our research requires continuation to verify the results. Subsequent research will concern (see also [3, p. 53]):

- testing an application for the ontology of ROI indicator with the participation of users with various education, in order to confirm the obtained data (using the same tasks and the same questionnaire),
- testing another tasks to be done in the application for the ontology of ROI indicator, in order to state whether tasks in the conducted research are comprehensible for people with non-computer education,
- testing another applications of ontologies created for other fields of the analysis of economical ratios, in order to verify the correctness of accomplishing the application for the ontology of ROI indicator (for which built ontology contains many more topics and relations).

In the next study, despite optimistic opinions, it will be necessary to plan longer training in searching information with the use of semantic network (30-40 minutes).

5 Summary and Future Work

In this article, we introduce the results of our initial research to verify the usability of applying the visualization of a semantic network. We focused on searching needed information related to the analysis of economic ratios. This research is the first formal user evaluation of an application related to ROI indicator. In this paper we presented the assumptions of research and we analyzed the obtained results.

The results of the research, despite their initial and fragmentary character, can be found as quite significant. They characterized the usability assessment of applying the visualization of the ontology of chosen economical ratio as interface user – system in searching information with regard to contextual connections. The research will be continued in order to verify using the visualization of a semantic network in the process of the analysis of economical ratios and to verify the use of the standard topic map for the representation of knowledge about economic measures. The study will use two developed ontologies for: ROI indicator and an early warning system for production enterprises. We will build the applications based on Ontopia (open source tool to create topic map). The research will be conducted in two ways: testing created topic maps applications for selected areas of economic analysis and studying the usability of visualizing semantic network to search information by using testing technique and heuristic evaluation.

Acknowledgements. This work was supported by the Polish Ministry of Science and Higher Education, grant N N111 284038.

References

1. Ahmed, K., Moore, G.: Applying Topic Maps to Applications. The Architecture Journal (January 2006), `http://msdn.microsoft.com/en-us/library/bb245661.aspx`
2. Arndt, H., Graubitz, H., Jacob, S.: Topic Map-based Indicator System for Environmental Management Systems (2008), `http://www.iai.fzk.de/Fachgruppe/GI/litArchive`

3. Dudycz, H.: Research on Usability of Visualization in Searching Economic Information in Topic Maps-Based Application for Return on Investment Indicator. In: Korczak J., Dudycz H., Dyczkowski M. (eds.) Advanced Information Technologies for Management – AITM 2011. Intelligent Technologies and Applications. Wroclaw University of Economics Research Papers no. 2o6, pp. 45–58, Wroclaw (2011)
4. Dudycz, H.: Visual Analysis of Economical Ratios in Du Pont Model Using Topic Maps. In: Polenakovik, R., Jovanovski, B., Velkovski, T. (eds.) Proceedings of the 4th International Conference for Entrepreneurs, Innovation and Regional Development – ICEIRD 2011, National Center for Development of Innovation and Entrepreneurial Learning, Ohrid-Skopje, Macedonia, pp. 277–284 (2011)
5. Eilrich, L., Andres, F., Sillaume, G., Backes, M.: Fostering User Experience in Order to Improve the Quality of a Digital Library. In: Workshop on Advanced Technologies for Digital Libraries (2009), `http://www.cacaoproject.eu/fileadmin/media/AT4DL/paper-04.pdf`
6. Garshol, L.M.: Metadata? Thesauri? Taxonomies? Topic Maps! Making Sense of It All (2004), `http://www.ontopia.net/topicmaps/materials/tm-vs-thesauri.html`
7. Grand, B.L., Soto, M.: Information Management – Topic Maps Visualization (2000), `http://www.gca.org/papers/xmleurope2000/pdf/s29-03.pdf`
8. Hunting, S., Park, J.: XML Topic Maps: Creating and Using Topic Maps for the Web. Addison-Wesley Professional (2002)
9. Jain, R.K., Kasana, R.S., Jain, S.: Visualization of Mined Pattern and Its Human Aspects. International Journal of Computer Science and Information Security 4(1,2), 48–54 (2009)
10. Koshman, S.: Testing User Interaction with a Prototype Visualization-Based Information Retrieval System. Journal of the American Society for Information Science and Technology 56(8), 824–833 (2005)
11. Lazar, J., Feng, J.H., Hochheiser, H.: Research Methods in Human-Computer Interaction. John Wiley & Sons Ltd. (2010)
12. Librelotto, G.R., Azevedo, R.P., Ramalho, J.C., Henriques, P.R.: Topic Maps Constraint Languages: Understanding and Comparing. International Journal of Reasoning-based Intelligent Systems 1(3/4), 173–181 (2009)
13. Pimentel, M.P., Suárez, J., Caparrini, F.S.: Topic Maps for Philological Analysis. In: Maicher, L., Garshol, L.M. (eds.) Linked Topic Maps. Fifth International Conference on Topic Maps Research and Applications, TMRA 2009, Leipziger Beiträge zur Informatik, Band XIX, Leipzig, pp. 29–39 (2009)
14. Sanin, C., Szczerbicki, E., Toro, C.: An OWL Ontology of Set of Experience Knowledge Structure. Journal of Universal Computer Science 13(2), 209–223 (2007)
15. Shneiderman, B.: Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison Wesley Longman, Addison Wesley Reading, Menlo Park (1998)
16. Sikorski, M.: User-System Interaction Design in IT Projects. Gdansk University of Technology, Gdansk (2012)
17. Tullis, T., Albert, B.: Measuring the Use Experience. Collecting, Analyzing, and Presenting Usability Metrics. Morgan Kaufmann Publishers (2008)
18. Ward, M., Grinstein, G., Keim, D.: Interactive Data Visualization. Foundations, Technique, and Applications. A.K. Peters, Ltd., Massachusetts (2010)

19. Won, S., Oh, S.G.: The Effects of Topic Map Components on Serendipitous Information Retrieval. In: Maicher, L., Garshol, L.M. (eds.) Subject-centric Computing. Fourth International Conference on Topic Maps Research and Applications, TMRA 2008, Leipziger Beiträge zur Informatik, Band XII, Leipzig, pp. 301–310 (2008)
20. Wurzer, J., Smolnik, S.: Towards an Automatic Semantic Integration of Information. In: Maicher, L., Garshol, L.M. (eds.) Subject-centric Computing. Fourth International Conference on Topic Maps Research and Applications, TMRA 2008, Leipziger Beiträge zur Informatik: Band XII, Leipzig, pp. 169–179 (2008)
21. Zhang, J.: Visualization for Information Retrieval. Springer, Heidelberg (2010)
22. Yi, M.: Information Organization and Retrieval Using a Topic Maps-Based Ontology: Results of a task-based evaluation. Journal of the American Society for Information Science and Technology 59(12), 1898–1911 (2008)

Appendix 1

Table 1. The heuristic evaluation of visualization in searching economic information for the ontology of ROI indicator

The criteria for assessment	Scale of usability interface evaluation	Breakdown of accomplishment of tasks (%)			
		All participants	Education		
			Computer	Computer science and econometrics	Non-computer
A. How would you rate the system in terms of visual clarity?	highly satisfactory	17	15	11	23
	satisfactory	29	25	**67**	8
	average	**31**	**40**	0	**38**
	unsatisfactory	21	20	22	23
	very unsatisfactory	2	0	0	8
B. How would you rate the system in terms of its functionality (in the context of searching information)?	highly satisfactory	7	10	11	0
	satisfactory	33	25	45	39
	average	14	10	22	15
	unsatisfactory	**40**	**45**	22	46
	very unsatisfactory	5	10	0	0
C. How would you rate the system in terms of flexibility of its structure and the presentation of information?	highly satisfactory	9	10	11	8
	satisfactory	**33**	20	**34**	**54**
	average	31	**45**	22	15
	unsatisfactory	26	25	33	23
	very unsatisfactory	0	0	0	0
D. How would you rate the way of searching information that bases on the visualization of semantic network?	highly satisfactory	7	10	11	0
	satisfactory	**36**	25	33	**54**
	average	24	30	11	23
	unsatisfactory	33	**35**	**45**	23
	very unsatisfactory	0	0	0	0

Table 2. The evaluation of the potential problems with usage of visualization of the semantic network in searching economic information

The list of the problems	Scale of the problem evaluation	Breakdown of accomplishment of tasks (%)			
		All participants	Education		
			Computer	Computer science and econometrics	Non-computer
1. Understanding how to navigate the OntoGraf	no problem	33	30	22	**46**
	a small problem	**57**	**65**	**67**	39
	an important problem	10	5	11	15
2. Understanding how to execute tasks	no problem	38	45	11	46
	a small problem	**55**	**50**	**78**	**46**
	an important problem	7	5	11	8
3. Understanding the relation between information on the screen and the executed operation	no problem	26	20	22	38
	a small problem	**57**	**75**	**45**	**39**
	an important problem	17	5	33	23
4. Finding necessary information	no problem	26	30	**45**	8
	a small problem	**60**	**60**	33	**77**
	an important problem	14	10	22	15
5. The difficulty in reading information on screen	no problem	**45**	**55**	33	38
	a small problem	43	35	**34**	**62**
	an important problem	12	10	33	0
6. Too many colors on the screen	no problem	**60**	**50**	44	**84**
	a small problem	24	25	**45**	8
	an important problem	16	25	11	8
7. The necessity to memorize too much information during executing of task	no problem	**55**	**55**	44	**61**
	a small problem	36	30	**56**	31
	an important problem	9	15	0	8

An Ontology-Based Event-Driven Architecture for Integrating Information, Processes and Services Applied to International Trade

Sietse Overbeek, Marijn Janssen, and Yao-Hua Tan

Faculty of Technology, Policy and Management,
Delft University of Technology, Jaffalaan 5, 2628 BX Delft, The Netherlands
{S.J.Overbeek,M.F.W.H.A.Janssen,Y.Tan}@tudelft.nl

Abstract. In global supply chains, many public and private organizations collaborate in order to succeed in transporting goods from the seller to the buyer. Given the dynamicity of global supply chains it is hard to predict which information is needed by whom at which point in time which oftentimes causes service delivery issues. Integrating relevant information, processes and services prevents deterioration in service provisioning caused by missing information required for processes that need to be executed to supply services. In this paper, an ontology-based event-driven architecture is described for integrating information, processes and services that acts as a mechanism to coordinate service delivery. The architecture is illustrated in the context of a global supply chain of plastic toys, where it is shown how the architecture enables the availability of valuable information based on events which positively influences the delivery of a barge planning service.

Keywords: CEP, EDA, global supply chain, ontology, service delivery.

1 Introduction

In global supply chains, many clusters of public and private organizations work together to make sure goods are safely and successfully transported from the selling party to the buying party. The composition of organizations that collaborate in these chains changes over time as new organizations might enter a chain while others might withdraw dependent of, for example, the goods that are traded in a specific trade lane. Flexible mechanisms are needed to coordinate service delivery in supply chains not only because of this kind of dynamicity, but also because complex client demands are subject to change in such chains and because supply and demand of services needs to be matched [1, 2]. Well-known mechanisms to coordinate service networks are service-oriented architectures (SOAs, see e.g. [1-3]) and event-driven architectures (EDAs, see e.g. [4-6]). Each organization that offers services in a supply chain can make its services accessible as Web services. Coupling services on the interface level is technically feasible, however, specific attention should be paid to issues of integrating the required information and underlying processes for service delivery as

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 143–158, 2012.

this could lead to deterioration in service provisioning when ignored [2]. Infrastructural support for process, information and service integration is one of the grand challenges as mentioned in the service-oriented computing research roadmap [1]. The ontological models and EDA that are presented in this paper will contribute to solving this challenge by integrating the architectural information, process, and service layers. Event-driven architectures (EDAs) enable the production, detection, consumption of, and reaction to business-critical events [4]. An event is defined as 'a significant change in state', while event consumers are those entities responsible for acting on these changes [7]. Hence, there is no need to draw out in detail the activities involved in a process which may even be impossible in complex scenarios. A motivating real-life example is used to illustrate how planning issues in a global supply chain of plastic toys are dealt with by the EDA presented in this paper.

The paper is structured as follows. Section 2 provides an overview of related work. The empirical motivation for the EDA is explained in section 3. Section 4 shows the high-level design of the architecture, which is further elaborated in section 5 by means of a formal event model and ontological models of the information, process, and service layers. The illustration of the architecture in the toys trade lane is presented in section 6. Finally, section 7 presents the conclusions and plans for future research.

2 Related Work

In this section, three categories of related work are discussed, which are: (1) studies related to the core technology of the EDA as part of this paper, which is complex event processing (CEP), (2) studies that combine event-driven concepts with service-oriented concepts, and (3) studies concerning models to integrate architectural layers. These three categories of related works will be subsequently dealt with in this order.

CEP is a core technology that is suitable for dealing with complex event streams related to information, processes and services [8]. This is done by processing and analysing multiple simple events from possibly distributed sources, with the objective of extracting semantically richer events from them in a timely, online fashion [9]. CEP effectively supports the implementation of 'sense and respond' behaviors, as it enables to extract meaningful events from raw data streams [9]. CEP is also part of the architecture as designed in this paper which is elaborated in section 4. An EDA for road traffic management systems is proposed in [4], for example, where CEP is used to extract meaningful events from raw measurements provided by traffic sensors and to deliver resulting events as input to traffic control systems for decision support. Experiment CEP techniques in a radio frequency identification (RFID)-enabled framework for managing hospital data for surgical procedures is presented in [10]. The used CEP engine models basic events and event patterns in hospitals for detecting medically significant events.

The added value of an event-driven architectural style next to SOAs is discussed in [5], as it is stated that dynamic interaction of service providing organizations in networks such as global supply chains trigger a considerable amount of meaningful state changes, i.e., events. The event-driven approach supplements the service-oriented approach by facilitating real-time event processing and distributed service coordination [5]. Another solution for extending the service-oriented architecture style with EDA concepts is presented in [6], showing that combining event-driven and service-oriented concepts does not only make sense in many business applications but also reduces development efforts of the architecture itself.

Merging, supplementing and combining architectural styles is still something different than integrating the different layers of a single architecture by means of ontological models, for example, as has been the case in our approach. Works like, e.g., [3, 4] do include event ontologies to classify event instances, but they are not specifically used to integrate architectural layers as is the case with the EDA in this paper. In earlier work we have used an ontology to generate both computer-interpretable and human-readable requirements for the execution of cross-organizational processes [11, 12]. In [13], ontologies are used to represent collaboration patterns in enterprises, as by using ontologies related concepts and interrelations can be effectively modeled. The presented ontology-based EDA is then used for event detection and reaction on those events in ongoing collaborations in enterprises, while in our case ontological models are used to describe and couple the information, process and service layers which enables to understand the information requirements that follow from executed processes to deliver services. As will be seen throughout the paper and, especially in section 6, integrating the three ontological layers as part of the EDA enables the availability of valuable information based on runtime events and in the mentioned plastic toys supply chain it contributes to solving service delivery issues in the context of supply chain planning.

3 Empirical Motivation

A toys trade lane that goes all the way from a toy factory in China via the Port of Shenzhen to the Port of Rotterdam, the Netherlands and ends in the hinterland in Venlo, the Netherlands serves as the context for the empirical motivation to design an ontology-based EDA for integrating information, processes and services. Toys that are ready after manufacturing are shipped to a warehouse nearby Shenzhen from where they are shipped to the Port of Shenzhen by road. Once loaded on a freighter, the cargo is shipped to the Port of Rotterdam where it is imported in the EU. From there, the cargo will be transported to a warehouse in Venlo. This trade lane is visualized in figure 1.

Fig. 1. A trade lane of plastic toys from China to the Netherlands

In such an international trade lane, all kinds of public and private organizations collaborate. These are, for example, providers of logistic services, regulatory authorities such as: customs, the tax administration, the food and consumer product safety authority, but also the consignor (aka the selling party) and the consignee of the toys (aka the buying party). These organizations exchange *information* with each other to make sure the toys are exchanged from the consignor to the consignee in a transparent, reliable and secure way. The Port of Rotterdam is an important decision point in the toys trade lane, because a decision has to be made whether the cargo of toys will be transported by road or by barge to Venlo. The logistic re-planning of the remainder of the cargo transportation route has to be done dynamically in Rotterdam based on the information at hand. Road transport is much more expensive than barge transport and causes considerable emissions of CO_2, but the advantage of road transport is that the destination can be reached faster when compared to barge transport. If an inland barge operator wants to make use of an online barge planning *service* at the Port of Rotterdam, it requires that the information *when* the cargo arrives at Rotterdam and on *which* freighter the cargo is stored needs to be known by the local *port community system* (PCS) two days before the scheduled barge departs from Rotterdam. A PCS is an entity that acts as a neutral hub which offers all kinds of online services to traders in the port such as the barge planning service. The PCS communicates with the terminal operator, which is the entity that loads the inland barges in Rotterdam with containers that have arrived from overseas. The mentioned information is required that early because the barge planning *process* has to be performed as efficiently as possible by only letting barges depart when they are fully loaded. This is why barges are loaded with mixed cargo, for example, a container on a barge may contain boxes with toys but also boxes with automotive parts that need to be dropped off in Venlo as well.

The problem in this outlined scenario is that it is difficult to get the information required for proper barge planning on time in Rotterdam. This is because it is hard to know the exact time a ship will arrive in Rotterdam. With an EDA, events that are triggered when a freighter with toy cargo leaves the Port of Shenzhen, changes its lane or arrives at the Port of Rotterdam can be processed and information can be derived from events that have occurred. A PCS can make use of this information to improve the delivery of their barge planning service. Currently, however, the required information for barge planning is often not available within the two days before the last inland barge departs that would arrive at the desired point in time in Venlo. This causes that often the only option left is to make use of costly and polluting road transport. The information on which freighter the cargo is stored and when cargo arrives would be available for the PCS when each state change of the cargo would trigger an event. Once the required information has been collected to supply the barge planning service, the PCS can perform the process to deliver the service to the barge operator. By integrating information for process execution with process and service descriptions through the EDA presented hereafter, the mentioned service provisioning issues can be remedied.

4 Integrating Information, Processes and Services

In the mentioned real-life example in international trade, the three layers of information, processes and services are of importance. First, information such as: where are the goods, what type of goods are transported, and an export declaration is needed by organizations involved in the toys trade lane. Second, processes are needed to understand who needs to do what in the context of the toys trade lane. For example, customs want to perform physical inspections on the toys to check for health or environmental risks and a terminal operator is responsible for transferring toy cargo from a freighter to an inland barge. Third, services are used to guide the goods from seller to buyer, such as goods ordering, customs clearing, barge planning, etc. An ontology-based event-driven architecture to integrate information, processes and services is shown in figure 2. The figure shows that the three layers are described by means of an ontology. This ontology is modeled as an Object-Role Modelling (ORM) [14] model and is explained in section 5. ORM is, like UML or ER, a conceptual modeling language that can be used for a variety of modeling purposes, such as the modeling of databases or ontologies. A specific advantage of using ORM is that objects are treated as concepts, which makes ORM immune to changes in the model that cause attributes to be remodeled as objects or relationships. The ontology describes the core concepts and relationships between those concepts on each layer and is used to determine how the three layers are interrelated. As can be concluded from figure 2, the information layer feeds the process layer, while the process layer feeds the service layer. Business processes require information for successful execution, otherwise a process cannot be fulfilled.

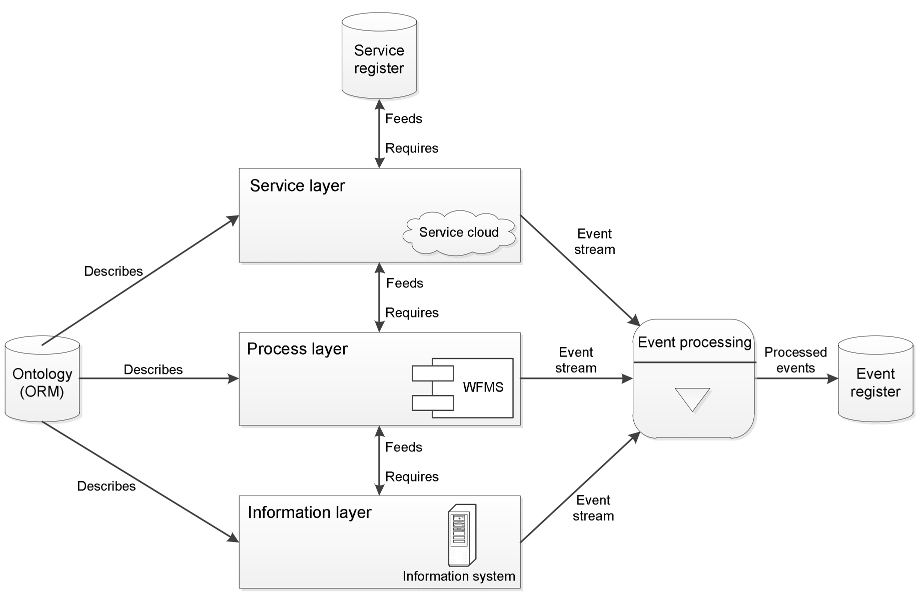


Fig. 2. An ontology-based EDA for integrating information, processes and services

The service layer requires processes to be executed, otherwise the services cannot be supplied to those actors in the toys trade lane that demand services. An example of such an actor is the buyer of the toys. Services are fed into a service register by the service providing organizations. The model shown in figure 2 does not have a separate software layer, as software applications are present in the three layers themselves. Computer-based support is delivered on the information layer in terms of enterprise information systems. Service providing organizations make use of dynamic workflow management systems (WFMSs) to execute, manage and monitor their business processes which is shown on the process layer. These WFMSs need to be dynamic because they need to be able to cope with runtime changes in business processes and process descriptions that are not always predefined. If services are offered in an online form then they are part of the cloud of Web services as is shown on the service layer.

The final parts of the architecture are the event processing part and the event register. Events from each layer are processed and then fed into an event register, which are next to the ontology parts of the architecture to integrate the three layers. Events are dealt with by means of complex event processing. In the next section we will zoom in on the parts of figure 2 that are related to event processing. Events are part of each layer, because the states of information, states of processes and service states change all the time when information is exchanged, business processes are executed and when services are delivered. A main principle of CEP is that events are not independent from each other, but correlated in space and time [4]. An example of correlation in space in the context of the toys trade lane can be events that are triggered at the port of departure in Shenzhen and at the port of arrival in Rotterdam. Those events can be correlated as it concerns toy cargo which could be tracked based on those correlated events. Event correlation in time means that events that are observed at one time instant are indicative of events observed at the next time instant or other future time instants. An example is event information showing that a shipment has departed from its port of departure or when an event that is triggered when a freighter changes its lane. Based on the estimated time of arrival (ETA) at the port of arrival, this event information can be indicative for the moment in time when events are observed that the shipment has indeed arrived at the port of arrival. These kind of correlations between events provide additional information that would remain hidden without time and space correlation of events. This event-driven approach is applicable to the toys trade lane, where supply chain planning in advance is considered inefficient and expensive as dynamic re-planning at the Port of Rotterdam is needed in order to reduce costs and increase efficiency. Having the ability to choose from different transport modalities at the node in Rotterdam means that costs can be lowered and efficiency can be increased which is made possible by applying the CEP mechanism instead of using top-down hierarchical planning.

Figure 3 shows the building blocks of CEP applied to the three layers that are found in figure 2. The event streams that are related to service provisioning, process execution and information exchange are processed by an event processing engine. This engine has to deal with a continuous flow of events. To process such a continuous flow, continuous queries are issued once and then run continuously over the event stream [15]. The event model is used to classify event instances based on their types and is further elaborated in section 5.1. The event processing rules define correlations between events in the form of event patterns and can be expressed by event processing languages based on event algebras or as SQL-like queries over event streams [15]. Events are stored in an event register after the events are processed by the engine, classified by means of the event model and correlated by means of the rules.

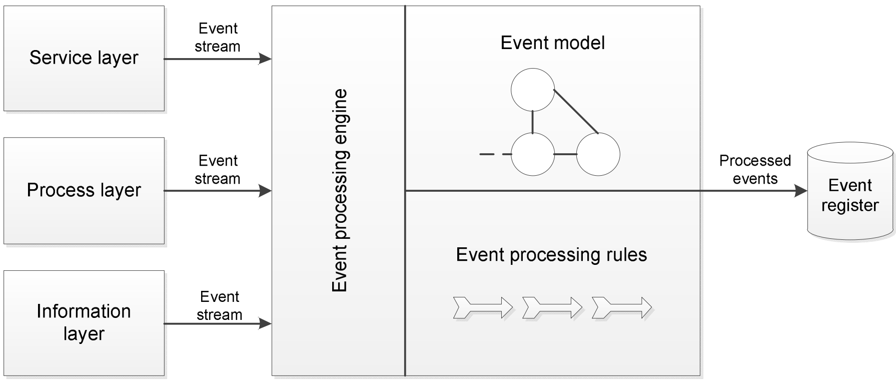


Fig. 3. Complex event processing building blocks related to the three layers, adapted from [4]

A software application based on CEP in the context of the toys trade lane is a dashboard on which it is shown that, for example, twenty containers arrive as one shipment from Shenzhen. Ten of these containers may contain toys that can be transported cheaply to the hinterland by means of barge transport and the other ten containers contain automotive parts that require a means of transport that is quicker than barge transport, as they need to arrive at a point in time at a consolidation warehouse in Venlo, the Netherlands that cannot be reached when a slower way of transport than road transport is chosen. In the next section, we will explain the event model as part of the architecture together with the ontological models.

5 Decomposed Ontological Models and an Event Model

The event model that is elaborated hereafter is comparable to the ontological models related to all three layers, but the event model serves a different purpose, which is the classification of event instances based on their types, event IDs, time- and date stamps, and their coordinates in space. Correlations can be made by the event processing engine based on this model and the event processing rules.

5.1 A Formal Event Model

Figure 4 shows the event model modeled with the ORM language, which is used to classify events so that they can be stored in an event register. In an ORM model, rounded rectangles represent object types (which are the counterparts of classes), while boxes represent relationships between object types. Bold arrows express specialization relationships. The event instance object type is obviously in the centre of the event model. This object type has the most relationships with the other object types in the model. An event instance can be classified as a certain event type. This can be formalized as follows: $\text{EType} : \mathcal{EV} \rightarrow \mathcal{ET}$. The expression $\text{EType}(e) = t$ shows that an event $e \in \mathcal{EV}$ is of the type $t \in \mathcal{ET}$, where $\mathcal{EV}$ is the set of events and $\mathcal{ET}$ is the set of event types. Figure 4 also shows three arrows that are drawn from the object type 'Event Type' to the object types 'Information Event', 'Process Event', and 'Service Event', which implies that each 'Information Event', 'Process Event', and 'Service Event' is also an 'Event Type'. Instances of information, process and service events are streamed from three different layers. The stream equation that captures this is modeled as follows: $\text{Stream} : \mathcal{EV} \rightarrow \mathcal{AL}$. The expression $\text{Stream}(e) = a$ shows that an event $e \in \mathcal{EV}$ is streamed from layer $a \in \mathcal{AL}$, where $\mathcal{AL}$ is the set of layers. However, there is a value constraint on the object type 'Architectural Layer' showing that only the values '`IL`', '`PL`', and '`SL`' are permitted. This means that it can only be modeled that events can be streamed from the information layer (`IL`), the process layer (`PL`), and the service layer (`SL`). These event subtypes form the relationship between the event model and the ontological models of the three architectural layers that are elaborated in the following sections. Because events can be correlated in space by the event processing engine, the ORM event model of figure 4 shows a visualization of the coordination equation. It is possible to reason about the spatial relationships between events if the coordinates are known where events take place. The three-dimensional coordinates of an event can be found by means of the coordinates equation: $\text{Coord} : \mathcal{EV} \rightarrow \mathbb{R} \times \mathbb{R} \times \mathbb{R}$. The coordinates are plotted on a three-dimensional Cartesian coordinate system, with an origin and axes *X*, *Y*, and *Z*.

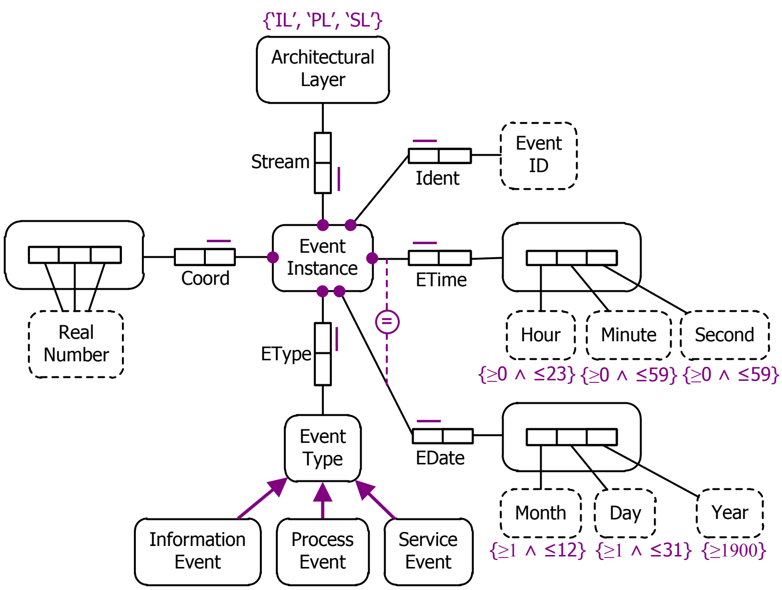


Fig. 4. An ORM-based event model

Events can also be correlated in time next to the correlation of events in space. This is why the event time and event date functions are shown in figure 4. The signatures of these equations are: $\mathrm{ETime}, \mathrm{EDate} : \mathcal{EV} \rightarrow \mathbb{N} \times \mathbb{N} \times \mathbb{N}$. For example, $\mathrm{ETime}(\mathrm{e}) = (22, 15, 42)$ expresses that an event instance took place at 22 hours, 15 minutes and 42 seconds. Figure 4 shows that a *set equality* constraint is added to the roles of the event time and date equations that are coupled to the 'Event Instance' object type. This constraint forces that time and date stamps should always be given together when some event instance occurs. Finally, each event instance has a unique ID. Therefore, the event identification equation is introduced to complete the event model: $\mathrm{Ident} : \mathcal{EV} \rightarrow \mathbb{N}$. Now that a formal event model has been explained as part of the architecture for integrating information, processes and services the ontology to describe the three layers will be elaborated.

5.2 Ontological Model of the Information Layer

The ontological model for describing the information layer is shown in figure 5. The model shows that the 'Agent State' object type plays a pivotal role in the ontological model and has the most relationships with other object types. Agents working for service providing organizations require information and, therefore, exchange information during the execution of business processes to deliver services. The basis for the ontological model of the information layer is the theory for demand and supply of information as presented in [16].

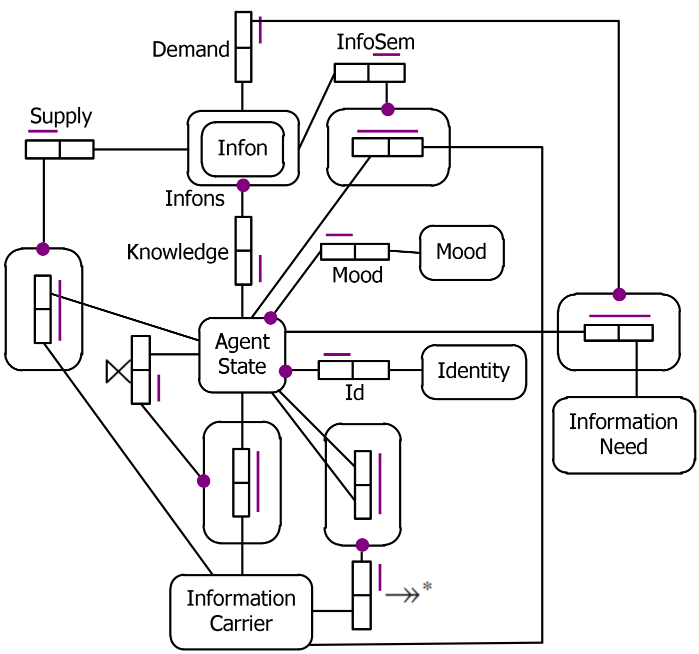


Fig. 5. An ontological model for describing the information layer

First of all, information needed by agents is provided on information *carriers*, such as Web pages, PDF documents, and e-mails. After the information retrieval from a carrier an agent is in a different state, i.e. that agent has absorbed more information than before. Each state belongs to a unique agent, determined by the identity function [16]: $\mathrm{Id}: \mathcal{AS} \rightarrow \mathcal{ID}$, where $\mathcal{AS}$ is the set of agent states and $\mathcal{ID}$ is the set of agent identities. This function shows that the identity of an agent $i \in \mathcal{ID}$ is determined by the agent state. When an agent in state a experiences an information carrier $c \in \mathcal{IC}$, where $\mathcal{IC}$ is the set of information carriers, then this agent will end up in a new state denoted as $a \ltimes c$. This is an expression of the following state change function: $\ltimes: \mathcal{AS} \times \mathcal{IC} \rightarrow \mathcal{AS}$. Note that for readability purposes the notation $a \ltimes c$ is a different notation than $\ltimes(a, c)$ but the semantics of both are the same. The knowledge that an agent has accrued is administrated by the knowledge function [16]: $\mathrm{Knowledge}: \mathcal{AS} \rightarrow \wp(\mathcal{IN})$. The expression $\mathrm{Knowledge}(a) = I$ shows that a set of information particles $\mathcal{IN}$, or *infons* [17] have been accrued by an agent in state a. For example, when a customs officer interprets an export declaration of toys he has acquired information from that declaration, causing him to change to another state.

The mood of an agent is also taken into consideration when designing the ontological model for the information layer, as this may influence how much knowledge is acquired from an information carrier. The moods of agents thus also influence the results of information exchange. The mood function is modeled as follows [16]: $\mathrm{Mood}: \mathcal{AS} \rightarrow \mathcal{MO}$. The mood $m \in \mathcal{MO}$ of an agent in state a can be expressed as $\mathrm{Mood}(a) = m$. Assume that a customs officer in state a interprets information from an export declaration form a couple of times. This form is an information carrier $c \in \mathcal{IC}$. Further assume the following: $\{\mathtt{interested}, \mathtt{informed}, \mathtt{bored}\} \subseteq \mathcal{MO}$. Before interpreting the export declaration the customs officer is interested: $\mathrm{Mood}(a) = \mathtt{interested}$. After reading the export declaration form the customs officer is informed: $\mathrm{Mood}(a) = \mathtt{informed}$. After reading the form another time the officer is bored: $\mathrm{Mood}(a) = \mathtt{bored}$. The customs officer will most probably hardly consume any new information after reading it for a third time, making him bored.

Agent states can also be ordered by means of the experience operator: $\twoheadrightarrow^* : \mathcal{AS} \times \mathcal{AS} \rightarrow \mathcal{IC}$. This means that $a_1 \twoheadrightarrow^* a_2$ can be interpreted as an agent in state $a_1 \in \mathcal{AS}$ ends up in state $a_2 \in \mathcal{AS}$ after experiencing an information carrier [16]. Next to the agents that have a certain role in the context of information exchange, the information itself is transported from an information supplier to an information requester. The supply of information is modeled as follows: $\mathrm{Supply}: \mathcal{AS} \times \mathcal{IC} \rightarrow \wp(\mathcal{IN})$. This function implies that an agent in state a supplies information $I \subseteq \mathcal{IN}$, which is carried by an information carrier c. The potential information that a carrier may provide to an agent can be expressed by using the information semantics function [16]: $\mathrm{InfoSem}: \mathcal{IC} \times \mathcal{AS} \rightarrow \wp(\mathcal{IN})$. For example, the overall information content $J \subseteq \mathcal{IN}$ of an information carrier c for a given agent in a state a can be expressed as $\mathrm{InfoSem}(c, a) = J$. In this case, a customs officer that is interested in receiving information about toy cargo that he wants to inspect probably acquires more infons from an information carrier than a customs officer that is uninterested and does not want to receive information about a shipment. The information need that an agent has corresponds to a need for infons, which is modeled as follows: $\mathrm{Demand}: \mathcal{AS} \times \mathcal{IM} \rightarrow \wp(\mathcal{IN})$, where the set $\mathcal{IM}$ is the set of all information needs [16].

5.3 Ontological Model of the Process Layer

The ontological model of the process layer describes those concepts and relationships between the concepts that are of importance for using the information that has been exchanged in the underlying information layer for specifying the workflow that needs to be followed to realize service delivery. Figure 6 shows the ORM model for describing the process layer which visualizes the formalisms shown in this section. The process ontology in [18] forms a basis for the ontological model. First of all, process input is processed and transformed into output. Process input concerns the *infons* that are produced after process execution, which is done by one or more agents performing the process. The transformation of infons from service input to infons as service output is modeled as follows: $\text{Transform} \subseteq \mathcal{PR} \times \wp(\mathcal{IN}) \times \wp(\mathcal{IN})$. An expression like $(p, I_1, I_2) \in \text{Transform}$ shows that some process $p \in \mathcal{PR}$ transforms input $I_1 \subseteq \mathcal{IN}$ to output $I_2 \subseteq \mathcal{IN}$. The equation to express which agents perform which processes is modeled as follows: $\text{Perform} \subseteq \mathcal{AG} \times \mathcal{PR}$. For example, $(a, p) \in \text{Perform}$ shows that some agent $a \in \mathcal{AG}$ performs some process $p \in \mathcal{PR}$. The OWL-S specification [18] shows that there are three subtypes of the super type ‘Process’. These are ‘Atomic Process’, ‘Simple Process’ and ‘Composite Process’. An atomic process is one that has no internal structure. It corresponds to a single interchange of inputs and outputs. A composite process consists of a set of component processes linked together by control flow structures. The control flow is described using typical programming language or workflow constructs such as sequences, conditional branches, parallel branches and loops.

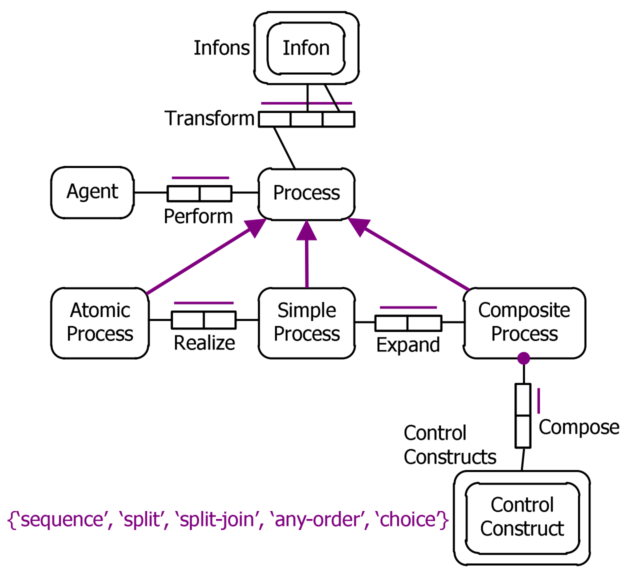


Fig. 6. An ORM-based ontological model for describing the process layer

A third type of process, the simple process, can be used to provide abstracted, non-invocable views of atomic or composite processes. A simple process is realized by an atomic process, while a simple process can be expanded to a composite process. A composite process on its turn is composed of the control constructs sequence, split, split-join, any-order and choice. The following equations and one function are introduced to formalize the relationships between the process subtypes: Realize $\subseteq \mathcal{AP} \times \mathcal{SR}$, Expand $\subseteq \mathcal{SR} \times \mathcal{CP}$ and Compose : $\mathcal{CP} \rightarrow \wp(\mathcal{CC})$. The set $\mathcal{AP}$ is the set of atomic processes, the set $\mathcal{SR}$ is the set of simple processes, the set $\mathcal{CP}$ is the set of composite processes, and the set $\mathcal{CC}$ is the set of control constructs. The expression $(p_1, p_2) \in$Realize shows that some atomic process $p_1 \in \mathcal{AP}$ realizes some simple process $p_2 \in \mathcal{SR}$, while a simple process $p_2 \in \mathcal{SR}$ that is expanded to a composite process $p_3 \in \mathcal{CP}$ is expressed as $(p_2, p_3) \in$ Expand. The expression Compose$(p_3) = C$ shows that a composite process is composed of control constructs $C \subseteq \mathcal{CC}$.

5.4 Ontological Model of the Service Layer

The ontological model of the service layer describes those concepts and relationships between the concepts that are of importance for the actual service delivery to clients by service providing organizations and it is shown in figure 7. The reference service model (RSM) [19] is suitable to serve as a basis because it aims to facilitate the semantic interlinking between services annotated using different semantic models and it accommodates bottom-up social annotation of services. The transformation of infons from service input to infons as service output is modeled as follows: Transform $\subseteq \mathcal{SE} \times \wp(\mathcal{IN}) \times \wp(\mathcal{IN})$, where $\mathcal{SE}$ is the set of services. This equation is almost identical to the transform equation used in the process model. A service is executed in a service *context*, which may differ at every new execution of a service. The context of a service is adapted by the circumstances of the service client [19].

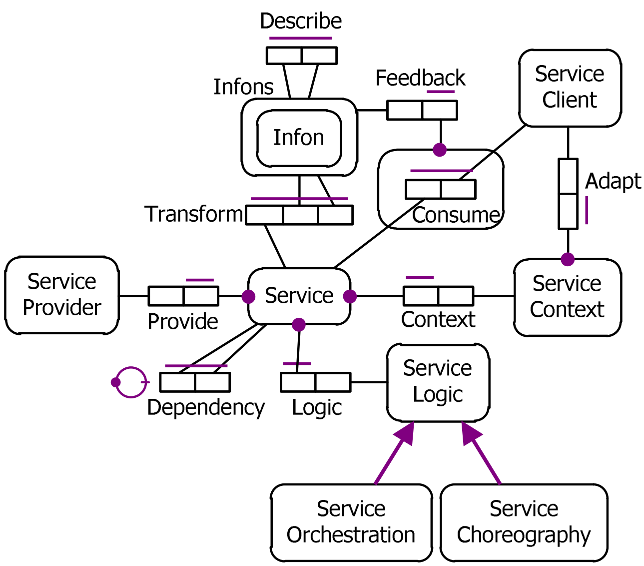


Fig. 7. An ORM-based ontological model for describing the service layer

For these reasons, the following two functions are introduced: $\text{Context} : \mathcal{SE} \rightarrow \mathcal{SC}$ and $\text{Adapt} : \mathcal{SC} \rightarrow \mathcal{CL}$. The context function is used to show that each service has a context, while the context adaptation function is used to show from which client the circumstances are adapted to form the context of a service. As the service *logic* concerns the business logic that is implemented by a service it determines both the input required by a service as well as the output produced by it. That each service implements a logic can be expressed by the logic function: $\text{Logic} : \mathcal{SE} \rightarrow \mathcal{SL}$. The interface of a service is described by defining its choreography and orchestration. Therefore, 'Service Orchestration' and 'Service Choreography' are introduced as two subtypes of the 'Service Logic' object type. The necessary input and output information for each service's capability are *described* in the choreography of the service. The orchestration reflects the *dependency* of a service with another service [19]. The symbol next to the dependency fact type in the figure depicts the ORM 'ring constraint' for *irreflexivity*. This means that in case of service orchestration some service cannot be dependent of itself. Each service is uniquely provided by some service provider and this is shown by the service provision function: $\text{Provide} : \mathcal{SE} \rightarrow \mathcal{SP}$. The consume equation is written as: $\text{Consume} \subseteq \mathcal{CL} \times \mathcal{SE}$ and is used to show that some client $c \in \mathcal{CL}$ consuming a service s is expressed as $(c, s) \in \text{Consume}$. Finally, a service provider can improve a service based on the feedback delivered by a client who has used the service. This is modeled as follows: $\text{Feedback} : \mathcal{CL} \times \mathcal{SE} \rightarrow \wp(\mathcal{IN})$. The situation in which a client c has consumed service s and provides feedback $F \subseteq \mathcal{IN}$ is expressed as $\text{Feedback}(c, s) = F$. With the description of the ontological model of the service layer complete, the international toy trade lane is used to illustrate the models.

6 Illustrating the Architecture in the Plastic Toys Trade Lane

The ontology-based event-driven architecture is illustrated in the context of the plastic toys trade lane to show how the utilization of the introduced formalisms can solve the barge planning issues raised in the trade lane. First, example events and expressions of equations in the formal event model are described. The expression $\text{EType}(e_1) = i$ shows that an event instance e_1 is an information event type i, which has been triggered when a freighter with toy cargo leaves the Port of Shenzhen. The event instance e_1 is streamed from the information layer, as it is an informational event, which is expressed as: $\text{Stream}(e_1) = \mathtt{IL}$. The coordinates in space where the event takes place are expressed by $\text{Coord}(e_1) = (51{,}12{,}11)$. These could be the GPS coordinates of the Port of Shenzhen.

The following two expressions $\text{ETime}(e_1) = (10{,}05{,}23)$ and $\text{EDate}(e_1) = (01{,}05{,}2012)$ show that a freighter with toy cargo leaves the Port of Shenzhen at 10:05:23 on 5 January 2012. Second, related expressions of the information layer can be introduced. Assume that a barge planning agent interprets a dashboard containing event-based information about freighters arriving at the Port of Rotterdam. When interpreting this information he ends up in a new state $a \ltimes c$. The dashboard is an information carrier c. After interpreting the information on the dashboard, the expression $\text{Knowledge}(a \ltimes c) = I_1$ shows that the barge planning agent has accrued information I_1. Assume that the set I_1 contains information about: (1) what the coordinates are of specific toy cargo that might be shipped by barge, (2) on which freighter this cargo is stored, and (3) the expected time of arrival of the freighter in Rotterdam.

When involving the process layer model it can be determined how to view the toys trade lane in terms of processes. The first path of the trajectory from Shenzhen to Rotterdam can be viewed as a simple process p_1. In that part of the trajectory no complex dynamic re-planning has yet to be done, which causes that this first part can be depicted as a *black box* process to avoid overspecification. Once the cargo arrives in Rotterdam, the simple process can be expanded to a composite process which is expressed as $(p_1, p_2) \in$ Expand. It is possible to decompose process p_2 into one of two sub-processes $p_3$and $p_4$at the point where the toy cargo arrives in Rotterdam, because at that point it has to be decided if the trajectory from Rotterdam to Venlo is traversed by road or barge. By using control constructs such as `If-Then-Else` a choice can be made which sub-process has to be performed to continue the transport from Rotterdam. An `If-Then-Else` control construct occurs at the point in the process where it is identified that *if* the current ETA of the vessel arriving in Rotterdam is 2 days or less before the point in time when the last inland barge that would arrive on time in Venlo departs from Rotterdam, *then* road planning should be performed, *else* barge planning should be performed. As we know from the expressions as part of the information layer, choosing between road or barge is dependent whether required information from the event dashboard has been accrued.

Finally, meaningful expressions from the service layer model are shown. A barge planning service s tranforms input I_1 to output I_2, that is expressed as $(s, I_1, I_2) \in$ Transform. Recall that the information contained in I_1 is the barge planning information that has been acquired by the barge planning agent as mentioned above. The set I_2 contains information about (1) on which inland barge the toy cargo will be stored, (2) when it departs to Venlo, and (3) when it is expected to arrive there. Service s is a Web service and the output I_2 is supplied online to the terminal operator and the inland barge operator. The expression $\text{Logic}(s) = o$ shows that logic o is the service orchestration of the barge planning service. The orchestration shows that this service is dependent of the 'freight announcement service' u, as it is required for each freighter that arrives in Rotterdam to announce itself. This is expressed as: $\texttt{Dependency}(s, u)$. The service choreography of the barge planning service describes that the input of the service is I_1 and the output is I_2, which is expressed as: $\texttt{Describe}(I_1, I_2)$. The illustration in this section shows that triggered events are catalysts for generating a source of information, which, once made visible on a dashboard for in this case the PCS as the service provider, is used to improve the level of their service provisioning. Based on events, the service provider can get information about what the coordinates are of specific toy cargo that might be shipped by barge, on which freighter this cargo is stored, and the expected time of arrival of the freighter in Rotterdam. The EDA as elaborated in this paper shows that this architecture enables the availability of valuable information based on events. We have shown some example events, but it is conceivable that all kinds of other events are triggered when the states of the tracked toy cargo change when the cargo goes from China to the Netherlands. These events then provide additional valuable information for exchange between agents, which enables the execution of business processes to deliver logistic services, such as the barge planning service.

7 Conclusions and Future Research

Organizations that collaborate with each other in global supply chains are dependent of the information that is available to supply their services to demanding parties. Yet, the dynamic nature of global supply chains complicates the exchange of information as each time different organizations might be involved. In this paper, an ontology-based event-driven architecture is introduced that integrates information, processes, and services. This architecture can be used as a mechanism to coordinate service delivery in the context of, for example, a certain global supply chain where public and private organizations collaborate. It is illustrated by means of a real-life example in the context of a plastic toys trade lane that events are triggered by changes in the state of toy cargo. These events provide information to service providing organizations, for example, a port community system (PCS) uses information to supply a barge planning service for terminal operators and barge operators. A PCS is an entity that acts as a neutral hub which offers all kinds of online services to traders in a port.

For proper barge planning in the toys trade lane, the PCS needs to know when the toy cargo arrives at the Port of Rotterdam, the Netherlands, on which freighter the cargo is stored and when the cargo needs to be in some consolidation warehouse in Venlo. The PCS needs to have this information two days before the last inland barge departs that would arrive at the desired point in time in Venlo. If this information is not available on time, which happens frequently in the contemporary situation, then the only option to transport the toy cargo on the remainder of the trade lane from Rotterdam to Venlo is by road. This is much more expensive and polluting than barge transport. The illustration shows that the availability of information based on events positively influences the delivery of services. The architecture couples the information required to conduct a barge planning process with the delivery of the results of that process as a service to demanding parties.

Future research is concentrated on improving the architecture design, especially by further analyzing the exact ways how the three layers are coupled, which will probably lead to more insights in the relationships between which information is required for which processes to further enhance service delivery. Finally, the development of a prototype dashboard application that shows how information is derived from triggered events in the context of the plastic toys trade lane will also be researched. With this exercise, it is aimed to contribute to solving current planning issues in the toys trade lane by means of exploiting the architecture.

Acknowledgements. This paper results from the CASSANDRA project, which is funded by the 7th Framework Programme of the European Commission (FP7; SEC-2010.3.2-1) under grant agreement no. 261795. The CASSANDRA project addresses the visibility needs of business and government in the international flow of containerized cargo. It does so by developing a data-sharing concept that allows an extended assessment of risks by both business and government, thereby enabling enhanced supply chain visibility and cost-efficient and effective security enhancement. Ideas and opinions expressed by the authors do not necessarily represent those of all partners.

References

1. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: Service-oriented computing: A research roadmap. International Journal of Cooperative Information Systems 17, 223–255 (2008)
2. Chung, J.-Y., Chao, K.-M.: A view on service-oriented architecture. Service Oriented Computing and Applications 1, 93–95 (2007)
3. Yuan, S.-T., Lu, M.-R.: An value-centric event driven model and architecture: A case study of adaptive complement of SOA for distributed care service delivery. Expert Systems with Applications 36, 3671–3694 (2009)
4. Dunkel, J., Fernández, A., Ortiz, R., Ossowski, S.: Event-driven architecture for decision support in traffic management systems. Expert Systems with Applications 38, 6530–6539 (2011)
5. Kong, J., Jung, J.-Y., Park, J.: Event-driven service coordination for business process integration in ubiquitous enterprises. Computers & Industrial Engineering 57, 14–26 (2009)
6. Juric, M.: WSDL and BPEL extensions for Event Driven Architecture. Information and Software Technology 52, 1023–1043 (2010)
7. Scheer, A.-W.: ARIS - Business Process Modeling. Springer, Berlin (2000)
8. Luckham, D.: The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. Addison-Wesley, Boston (2002)
9. Zappia, I., Paganelli, F., Parlanti, D.: A lightweight and extensible Complex Event Processing system for sense and respond applications. Expert Systems with Applications (in press)
10. Yao, W., Chu, C.H., Li, Z.: Leveraging complex event processing for smart hospitals using RFID. Journal of Network and Computer Applications 34, 799–810 (2011)
11. Overbeek, S.J., Janssen, M.F.W.H.A., van Bommel, P.: A standard language for service delivery: enabling understanding among stakeholders. Computer Standards & Interfaces 34, 355–366 (2012)
12. Overbeek, S.J., Janssen, M.F.W.H.A., van Bommel, P.: Designing, formalizing and evaluating a flexible architecture for integrated service delivery: combining event-driven and service-oriented architectures. Service Oriented Computing and Applications (in press)
13. Papageorgiou, N., Verginadis, Y., Apostolou, D., Mentzas, G.: Event-driven adaptive collaboration using semantically-enriched patterns. Expert Systems with Applications 38, 15409–15424 (2011)
14. Halpin, T.: Information Modeling and Relational Databases: from Conceptual Analysis to Logical Design. Morgan Kaufmann, San Mateo (2001)
15. Arasu, A., Babu, S., Widom, J.: The CQL Continuous Query Language: Semantic Foundations and Query Execution. VLDB Journal 15, 121–142 (2006)
16. van Bommel, P., Proper, E., van der Weide, T.: Information coverage in advisory brokers. International Journal of Intelligent Systems 22, 1155–1188 (2007)
17. Devlin, K.: Infons and types in an information-based logic. In: Cooper, R., Mukai, K., Perry, J. (eds.) Situation Theory and its Applications, vol. 1, pp. 79–96. Center for the Study of Language and Information, Stanford (1990)
18. Martin, D., Burstein, M., McDermott, D., McIlraith, S., Paolucci, M., Sycara, K., McGuinness, D., Sirin, E., Srinivasan, N.: Bringing Semantics to Web Services with OWL-S. World Wide Web 10, 243–277 (2007)
19. Loutas, N., Peristeras, V., Tarabanis, K.: Towards a reference service model for the Web of Services. Data & Knowledge Engineering 70, 753–774 (2011)

Towards Requirements and Architecture Co-evolution

João Pimentel[1], Jaelson Castro[1], Emanuel Santos[1], and Anthony Finkelstein[2]

[1] Universidade Federal de Pernambuco - UFPE, Centro de Informática, Recife, Brazil
{jhcp,jbc,ebs}@cin.ufpe.br
[2] University College London - UCL, Department of Computer Science, United Kingdom
a.finkelstein@ucl.ac.uk

Abstract. The relationship between requirements and architectures is an important research field on software engineering. One of its challenges is to provide proper support for their co-evolution, i.e., how to assess the mutual impact of requirements and architecture changes on each other, as well as how to react to these changes in order to prevent misalignment between them. We advocate the use of a single goal model to express both requirements and architectural concerns. In this paper we put forward an approach for requirements and architecture co-evolution considering such a model. Moreover, we outline the reasoning required in order to support forward and backward co-evolution of service oriented systems.

Keywords: System architecture, Requirements engineering, Software evolution, Self-Adaptation, Service-oriented architectures, Autonomics.

1 Introduction

Software evolution has become a key research area in software engineering [10]. Software artifacts and systems are subject to many kinds of changes, which range from technical adjustments due to rapidly evolving technological platforms, to modifications in the software systems themselves required by the natural evolution of the businesses and requirements supported by them. These modifications include changes at all levels, from requirements through architecture and design, as well as source code, documentation and test suites. For consistent evolution, all models and artifacts should remain aligned as the software evolves.

For instance, whenever requirements change we need to assess whether the current architectural configuration continues to meet the stakeholders' requirements. Similarly, if the properties of components in an architectural model are modified we need to analyze if these changes affect requirements satisfaction. In both cases, when there is a mismatch between architecture and requirements, an architectural reconfiguration may be considered. This is particularly relevant in the case of services, since they are very dynamic and may change in several ways (functional upgrades, varying quality-of-service, withdrawn, so on and so forth). In contrast, when a traditional COTS component evolves, the system using it may continue to use an older version of that component. In the services case, evolution cannot be prevented.

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 159–170, 2012.

Since the abstraction level of software architecture is adequate for identifying and analyzing the ramifications of changes [14], it could be one of the software evolution pillars. Certainly, it is of paramount importance to identify when and why to perform changes, as well as to assess their impacts [4]. Recent advances in the Requirements Engineering and Software Architecture fields include methods and techniques to address the evolution, in isolation, of requirements and of architectural models. However, there is a lack of proposals for tackling the co-evolution of requirements and architecture.

In fact, the line that separates requirements from architecture is a blurred one, as argued in [5]. The Twin Peaks model highlights the intertwined characteristics of requirements and architectural models [27]. Requirements lie in the problem space, whilst architectures are part of the solution space. Thus, investigating how to define an architecture (solution) that satisfies the requirements (problem) is a key challenge in software engineering. Moreover, it is important to maintain this satisfaction throughout a system lifecycle [7].

In this paper we present a novel approach for dealing with requirements and architecture co-evolution. We define the co-evolution problem as the problem of assessing the impact of both requirements and architectural changes and responding to these changes.

The remainder of this paper is structured as follows. In Section 2 we present the case study used throughout the paper. In section 3 we describe the approach itself. Section 4 discusses related works. Lastly, Section 5 concludes the paper with a critical discussion of our proposed approach and indicates points for improvement.

2 Case Study

In this work we are expressing requirements using the *i** Framework [39]. It defines goal-based models to describe both the system and its environment in terms of intentional dependencies among strategic actors. The actors are refined using four kinds of elements: goal, softgoal, task and resource. Goals represent the actors' intentions, needs or objectives to fulfill its role within the environment in which they operate. Softgoals also represent the strategic interests of the actors, but in this case these interests are of subjective nature – it is generally used to express non-functional requirements. The tasks represent a way to perform some activity, i.e., they show how to perform some action to obtain the satisfaction of a goal or softgoal. The resources represent data, information or a physical resource that an actor may provide or receive. These elements are linked together within the actor boundaries using means-end, task-decomposition, and contribution links. The means-end links define which alternative tasks (means) may be performed in order to achieve a given goal (end). The task-decomposition links describe what should be done to perform a certain task (i.e., its sub-tasks). Finally, the contribution links suggest how a task can contribute (positively or negatively) to satisfy a softgoal. These contributions allow the selection of alternative tasks driven by the satisfaction of softgoals.

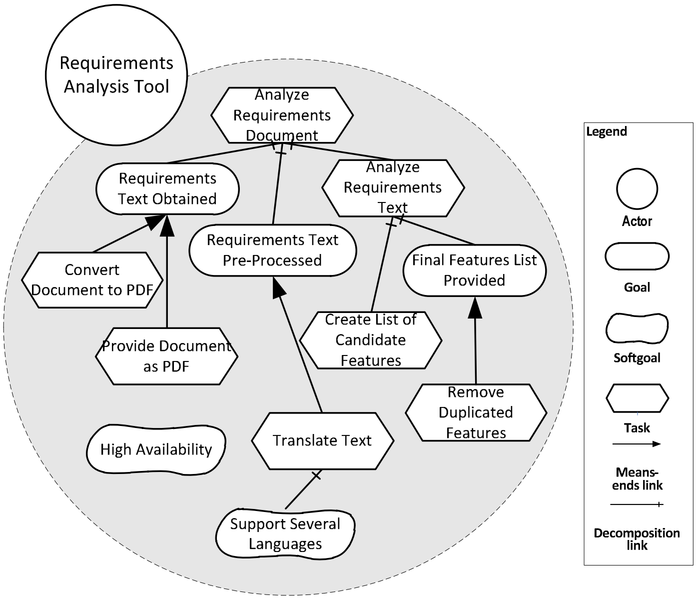


Fig. 1. The requirements model for the Requirements Analysis Tool

Fig. 1 presents the original requirements model of our system, which is a Requirements Analysis Tool. It is a web-based system that analyzes a textual requirements document and generates a list of candidate features. Thus, the main task of this system is to analyze a requirements document. In order to do so, it will need to obtain a requirements document, which will be provided by a user (here we are omitting dependency links to the system users). The user can either provide the document as a PDF file, or provide it in any usual file format (such as word processing documents and spreadsheets), which will be converted to PDF for our processing.

A common constraint on natural language text analysis is that it is highly dependent on the language being used. In order to enable the analysis of requirements documents in a wide range of languages, we decided to incorporate the functionality of translating the document to a reference language (the alternative would be to adapt the analysis algorithm for each language that we want to support). In order to reach a large user base worldwide, we defined that this translation must support several languages. The requirements analysis itself consists of creating a list of candidate features, and finally providing a consolidated list by removing duplicated features. Lastly, the *High Availability* non-functional requirement (softgoal) is important for our system, since it will be accessed anywhere, any time of the day. Please note that we have not yet defined how to satisfy this softgoal, since we have not taken any architectural decision yet. Alternatively, we could have already modeled all the different ways of satisfying this requirement – later we would only select which ones to use.

In Fig. 2 we present a possible structural architecture for the Requirements Analysis tool. In this architecture we rely on two kinds of services: Document Converter, which are services that provide file type conversion of documents; and Text Translator services, which are able to translate a given text. On the service consumer side, the client-server style was selected because it is well suited for web-based systems.

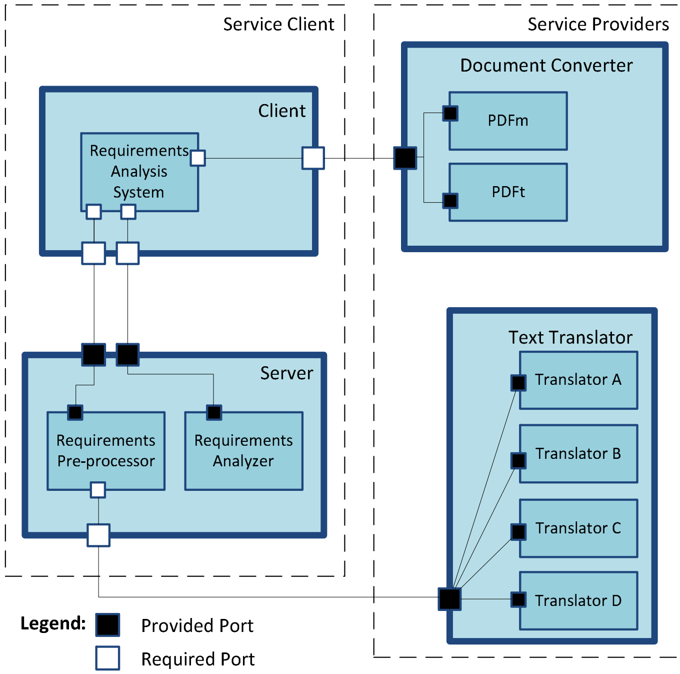


Fig. 2. Architectural model for the Requirements Analysis Tool, in Acme

3 Forward and Backward Evolution of Service-Oriented Systems

When dealing with requirements and architecture co-evolution two situations may arise. On one hand, changes in the system requirements may happen (this includes the system context, stakeholders' attitudes and quality constraints). In this case, some analysis is required to assess if these changes call for a reconfiguration, and whether there is some reconfiguration that satisfies the new requirements. We call this forward evolution, since it is from requirements to architecture.

On the other hand, there may also be changes in the architecture itself. For instance, the performance of a component may degrade. Thus, it is now required to check whether these architectural changes prevent requirements satisfaction. If this is the case, and this failure is unacceptable, it is necessary to attempt to identify a possible architectural reconfiguration that improves the level of the satisfaction of the requirements. However, if it is not feasible to reconfigure it, the system administrator could be prompted to either relax the affected requirements, or to perform offline evolution. This we call backward evolution – from architecture to requirements.

We propose to tackle the co-evolution problem by converging requirements and architecture models, i.e., working with architectural models that also contain requirements information. By doing so, we are able to perform the co-evolution reasoning in a single model. Moreover, this reduces the overhead of maintaining traceability between requirements models and architecture models.

In order to do so, we use a conventional requirements modeling notation to represent architectures – namely, *i**. This was preferred over modifying an Architectural Description Language (ADL) because *(i)* we did not find an architectural language expressive enough for presenting requirements; *(ii)* by using the same framework for both RE and architecture we can have a seamless approach to go from requirements to architecture; and *(iii) i** showed to be a suitable notation for expressing architectures.

Despite being an organizational modeling notation, *i** has shown to be particularly adequate for requirements modeling [39]. Recent works also showed that it is reasonably suitable for architectural modeling [16] [30]. More specifically, it has been used to model information services [26]. In [16] there are arguments in favor of using *i** extended models for architectural modeling. It is claimed that it can be used to describe main architectural concepts, such as components, connectors, constraints, non-functional properties and evolution. Moreover, *i** has suitable composition, abstraction and analysis mechanisms. However, it lacks proper support to promote reusability and heterogeneity, as well as it lacks proper support for configuring the models.

It is claimed that software architecture describes a system in a high-abstraction level, defining its components, the interaction among these components, their attributes and their functionalities [37]. Fig. 3 presents our approach for expressing service-oriented architectures using *i**, in the context of our case study. Here, conventional components were mapped onto *i** actors, service categories onto roles and the services themselves onto agents. A service category is a general definition of the service that is required, while an agent is a specific service that plays the role defined by a service category. E.g., Text Translator is a service category, whilst Microsoft Translator is a particular service of that category. With this mapping we are able to express the requirements related to each component of the architecture. Lastly, connectors are represented by dependencies. This allows expressing what is expected from a component (*dependum*), why is it expected (from the *depender*'s model) and how is it going to be provided (from the *dependee*'s model).

In Table 1 we present a summary of this mapping, considering the five major architectural elements [38]. However, note that the rationale, i.e., the information that explains the architectural decisions taken, cannot be properly captured by *i** elements. This is also the case for the majority of architectural modeling notations, where other artifacts are required to document the architectural diagrams [8].

Table 1. Mapping of architectural elements onto *i**

Architectural Element	*i** Element
Component	Actor, role, agent
Connector	Dependency links
Interface	Implicitly defined by the source and target elements of dependency links
Configuration	The graph itself
Rationale	Partially defined by internal elements (goals, softgoals, tasks, resources and their relationship)

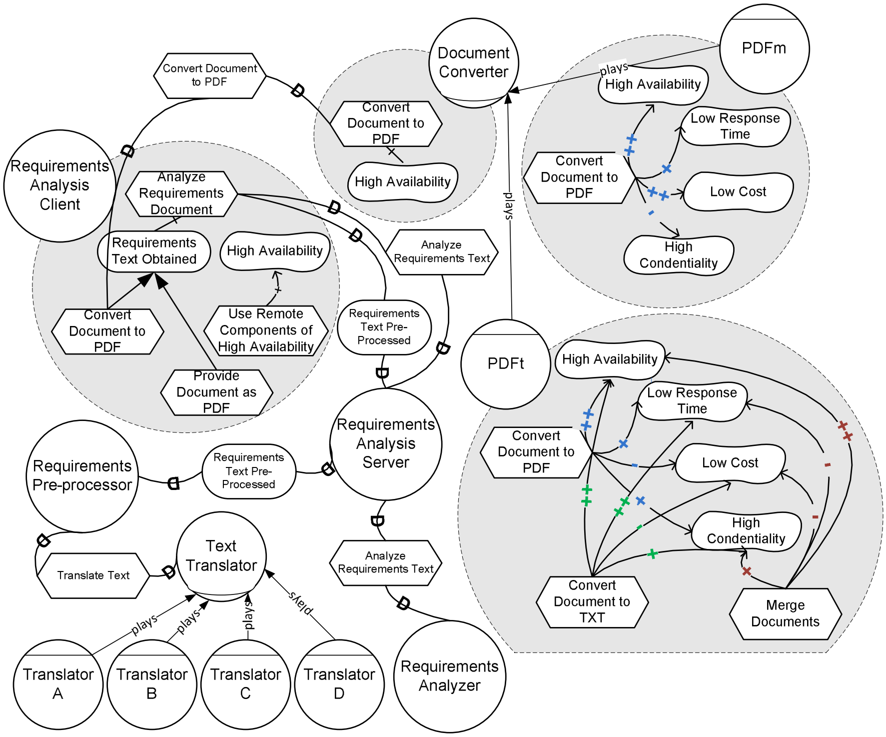


Fig. 3. Architectural model of the Requirements Analysis tool, using *i**. It replaces the former model presented in Fig. 2.

Another motivation for the use of *i** as an architectural description language is the current set of available reasoning mechanisms. Particularly relevant to our approach is the evaluation of the softgoals' satisfiability [15][19]. This allows selecting the best alternative to achieve a goal, considering the contributions of each alternative to the softgoals of interest (top-down reasoning). We can also assess whether a given alternative properly satisfies the selected softgoals.

3.1 Forward Evolution

Our concern here is to handle requirements changes related to the information services being used by the proposed system. These may come in two ways: a functional change, i.e., we want the service to satisfy a different goal or task; or a non-functional change, i.e., we define different quality constraints on how the service is supposed to support its goals or tasks.

When there are requirements changes, we are capable of checking whether the information service currently selected can satisfy the new requirements. For instance, consider that we are interested in the *Document Converter* service category (as described in Fig. 3) and that the *PDFm* service is currently selected. Several queries can be performed:

Query 1: Can PDFm support Convert Document to PDF with high Availability?
According to Fig 3. the service is highly available. However, if after deployment we notice that requirements documents are sometimes split in several documents according to some criteria (such as by sub-system, by viewpoints, and so on), we may decide that we also need the capability of merging documents. Thus, we can now pose a new question to check if *PDFm* provides this functionality as well, which is expressed in Query 2.

Query 2: Can PDFm support Convert Document to PDF and Merge Documents?
Since *PDFm* is unable to perform the *Merge Documents* task (Fig. 3), the answer to Query 2 is negative. Thus, we may ask the same question to other services of the same category. If one is found (eg. *PDFt*), we could then perform the required architectural reconfiguration, i.e., use *PDFt* instead of *PDFm*.

The same reasoning presented so far can be performed with softgoals as well. For instance, we may decide to go only for PDF conversion, as long as it is performed at low cost. In order to check whether *PDFm* satisfy this new requirement, we may ask Query 3:

Query 3: Can PDFm support Convert Document to PDF with low cost?
The assessment of softgoal satisfaction is trivial in the *PDFm* model (Fig. 3): since there is only one contribution link towards *Low Cost*, and it is a ++ contribution. Hence, the *Low Cost* softgoal is satisfied. For more complex cases, with different contribution links, one may refer to [15][19].

3.2 Backward Evolution

On the other hand, when there is a change in properties of the information services, or when new candidate services are identified, a similar reasoning may follow. In this case, we may use a monitoring framework in order to retrieve updated information on the services' properties. For instance, the SALMon tool [28] is able to provide up-to-date data on web services' response time and availability, among others. With such monitoring capabilities, we are able to assess both at design time and at runtime the quality of the information services being used.

In our case study, consider that we require documents to be converted to PDF with a high availability service. Recall that before deployment we certified that the *PDFm* service satisfied this query; for this reason, we had selected it for use in our system. Nonetheless, after deployment we may have noticed a degradation of its availability. Thus, we need to check whether this service is still able to meet our requirements – i.e., we need to check if there is a possible solution for Query 1. I.e., the same query would be performed, now with the model updated for the new availability value. Provided that automated monitoring is available, this reasoning can be completely performed without human intervention. Hence, it is suitable for adaptive and autonomic systems, which could perform this checking at regular time intervals. Foresight methods may be used to define which requirements/architectural elements to monitor and at what time intervals [33].

If the experienced change prevents the information service from satisfying its related requirements, we may check if other services of the same service category are able to meet the requirements – in this case, *PDFt*.

3.3 Tolerance, Relaxation and Manual Evolution

On the last two sub-sections we outlined how we can reason to identify a mismatch between system requirements and information services. Moreover, we showed how we can attempt to solve this mismatch by searching for a possible reconfiguration. There are two questions that arise when performing this reasoning: (i) all mismatches must be solved or can we live with some mismatches? (ii) What happens when no reconfiguration is able to solve this mismatch?

In previous works we argued that not every failure requires compensation [32], acknowledging that distinct failures may have different impacts. In our specific case, we could rephrase it: not every mismatch between service and requirements (failure to satisfy requirements) demands a reconfiguration (compensation). We tackle this issue by allowing system administrators to define tolerance policies, using our previous framework [32]. Thus, the system administrator will be able to define different criteria to assess when a failure in satisfying requirements, resulting from architectural changes, needs to trigger a reconfiguration.

A main element in that framework is the tolerance policy, which consists of tolerance rules. These rules may be related to the system context as well as to a particular element of the goal model, or to the amount of failures that happened. With this framework we may decide to ignore when a service fails to support a given element of the architectural model (e.g., a quality constraint), in particular conditions. Hence, instead of searching for a possible reconfiguration, we will continue to use the same service.

Regarding the second question, if it is not possible to reconfigure to satisfy the evolved requirements, and assuming that the tolerance policy in place does not allow for that failure to be ignored, we envision two scenarios. On one hand, the system administrator will be prompted to adjust (relax) the current requirements so that there is at least one possible reconfiguration. Alternatively, manual (offline) evolution of the system may take place.

4 Related Works

The area of Software evolution has been largely studied. More recently, terms such as autonomics, self-adaptation and self-management have been used to describe systems that are able to dynamically evolve at runtime. Regarding requirements evolution, some approaches (such as Lapouchnian and Mylopoulos [23] and Ali et al. [2]) use the notion of context in order to identify which elements of the requirements model are active/enabled. Pimentel et al. builds on that to derive architectures that support requirements activation/deactivation [31]. Jian et al. [21] proposes mechanisms to allow the insertion of goals in the requirements model at runtime. The system is only capable to satisfy these new requirements by developing new modules for the system. Qureshi et al. [35] also allows the changing of goal models at runtime. It proposes a service lookup mechanism to identify services that may satisfy the new requirements. Franch et al. [13] define metrics related to non-functional requirements. In turn, the metrics are linked to service categories and services. Thus, its reconfiguration is based solely on the measurements of these metrics.

Similarly, there are several research works regarding architectural evolution. For instance, [9] defines adaptation conditions based on architectural properties as well as reconfiguration operations. Control events based on components' states are used in [3] to reconfigure the architecture connectors. Composition rules are deployed in [34] to dynamically define connections between components and aspects. Some of previous work also allowed the addition, removal, change and reconfiguration of components [32]. These works may have broader and more sophisticated mechanisms for architecture evolution than ours. However, they fail to relate this evolution to system requirements.

There are also works on the requirements and architecture relationship such as [12][17][20][22]. However, they do not tackle this problem as we do, i.e. by considering the architecture model as a refinement of the requirements model, along the lines of what was developed for problem frames in [17].

Pahl et al. [29] proposes to dynamically define service collaboration through a coordination space, on which a service consumer expresses its need for a particular kind of service, which may be satisfied by a service provider. However, it does not consider the other elements of the software architecture.

5 Discussion

Considering the architecture as a reification of the system under consideration, and the increasing adoption of technologies that facilitate architectural changes (such as the technologies behind web services and cloud computing), it is of utmost importance to understand and reason on the relationships between requirements and architectural models. This calls for systems that are able to react to changes in requirements (i.e. according to the stakeholders expectations), as well as dealing with changes in the system itself (architecture). Architectural changes include structural changes – e.g., replacing a component (due to a new update) – and properties changes – e.g., the performance of a component may have degraded.

Throughout this paper we outlined our approach for requirements and architecture co-evolution. The main contribution of this approach is that it provides proper reasoning to handle the reciprocal impact between requirements and architecture – i.e., the requirements and architecture co-evolution. In the particular case of information services we are able to assess the impact of such changes, as well as to identify whether and which reconfiguration is possible to react to a given change. Given that proper monitoring tools are set up, this reasoning can be used at runtime to enable autonomic and self-adaptive behaviors.

In order to provide such reasoning, we advocate the use of architectural models enriched with requirements data. Such model may be derived from requirements models through a series of decision/transformations steps (e.g., [6]). In this research we propose the use of *i** for both requirements and architecture modeling [16][30]. This approach has some drawbacks, as follows:

Lack of familiarity – software architectures are already accustomed to conventional ADL. Thus, the need to learn a new notation would be a barrier for the adoption of this approach.

Poor readability – architectural models may become more difficult to handle in our approach due to the additional requirements information.

Lack of tools – there are several tools to support conventional ADL – e.g., for automatic code generation. The lack of similar tools to support *i** may prevent some architects to adopt it.

The first two drawbacks may be mitigated by using the *i** information hiding mechanism, by improving the *i** visual syntax [25] and by using modularization mechanisms [1][11]. The third problem may be softened by developing new tools for *i**, or by translating the *i** models to a conventional ADL as described in [6][24].

We believe that our approach is suited not only to service-driven architectures, but also for any kind of architecture on which components have some degree of intentionality. This is the case for socio-technical systems, on which some responsibilities are delegated not only to software and hardware components, but also to organizations and human participants. This is also the case for agent-based systems, on which each agent has its own goals, that may or may not converge to the overall system goals.

A key limitation of our approach is that we only consider the structural view of the architecture. Thus, an important advance in future works would be to include other views [36], as well as behavioral concerns. It is also important to notice that we intend to support only the derivation of architectural models – detailed design, class diagrams, code, and so on, are currently out of the scope of our approach. Thus, we do not define some service details, such as protocols, publishing mechanisms, and so on.

A major improvement for our approach would be to use Artificial Intelligence (AI) mechanisms in order to enhance the reasoning here proposed – for instance, simulation techniques [18]. This would be an important step towards Intelligent Software Engineering, i.e., Software Engineering that makes use of AI techniques.

Acknowledgments. This work was partially supported by Brazilian agencies CAPES and CNPq.

References

1. Alencar, F., Castro, J., Lucena, M., Santos, E., Silva, C., Araújo, J., Moreira, A.: Towards modular *i** models. In: 25th ACM Symposium on Applied Computing, pp. 292–297 (2010)
2. Ali, R., Dalpiaz, F., Giorgini, P.: A Goal Modeling Framework for Self-Contextualizable Software. In: Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Ukor, R. (eds.) BMMDS 2009 and EMMSAD 2009. LNBIP, vol. 29, pp. 326–338. Springer, Heidelberg (2009)

3. Allen, R., Douence, R., Garlan, D.: Specifying and Analyzing Dynamic Software Architectures. In: Astesiano, E. (ed.) ETAPS 1998 and FASE 1998. LNCS, vol. 1382, pp. 21–37. Springer, Heidelberg (1998)
4. Andersson, J.: Issues in Dynamic Software Architectures (2000)
5. Boer, R., Vliet, H.: On the Similarity between Requirements and Architecture. The Journal of Systems and Software 82(3), 544–550 (2009)
6. Castro, J., Lucena, M., Silva, C., Alencar, F., Santos, E., Pimentel, J.: Changing attitudes towards the generation of architectural models. Journal of Systems and Software 85(3), 463–479 (2012)
7. Cleland-Huang, J., Marrero, W., Berenbach, B.: Goal Centric Traceability: Using Virtual Plumblines to Maintain Critical Systemic Qualities. IEEE Transactions on Software Engineering 34(5) (2008)
8. Dermeval, D., Soares, M., Alencar, F., Santos, E., Pimentel, J., Castro, J., Lucena, M., Silva, C., Souza, C.: Towards an *i**-based Architecture Derivation Approach. In: Proceedings of the 5th International *i** Workshop, Italy, pp. 66–71 (2011)
9. Dowling, J., Cahill, V.: The K-Component Architecture Meta-model for Self-Adaptive Software. In: Matsuoka, S. (ed.) Reflection 2001. LNCS, vol. 2192, pp. 81–88. Springer, Heidelberg (2001)
10. Fernandez-Ramil, J., Perry, D., Madhavji, N.H. (eds.): Software Evolution and Feedback: Theory and Practice. Wiley, Chichester (2006)
11. Franch, X.: Incorporating Modules into the *i** Framework. In: Pernici, B. (ed.) CAiSE 2010. LNCS, vol. 6051, pp. 439–454. Springer, Heidelberg (2010)
12. Franch, X., Botella, P.: Putting Non-functional Requirements into Software Architecture. In: 9th International Workshop on Software Specification and Design (1998)
13. Franch, X., Grünbacher, P., Oriol, M., Burgstaller, B., Dhungana, D., López, L., Marco, J., Pimentel, J.: Goal-driven Adaptation of Service-Based Systems from Runtime Monitoring Data. In: 5th IEEE Workshop on Requirements Engineering for Services, Germany (2011)
14. Garlan, D., Perry, D.: Introduction to the Special Issue on Software Architecture. Journal IEEE Transactions on Software Engineering 21(4) (1995)
15. Giorgini, P., Mylopoulos, J., Nicciarelli, E., Sebastiani, R.: Formal Reasoning Techniques for Goal Models. In: 21st International Conference on Conceptual Modeling (2002)
16. Grau, G., Franch, X.: On the Adequacy of *i** Models for Representing and Analyzing Software Architectures. In: Hainaut, J.-L., Rundensteiner, E.A., Kirchberg, M., Bertolotto, M., Brochhausen, M., Chen, Y.-P.P., Cherfi, S.S.-S., Doerr, M., Han, H., Hartmann, S., Parsons, J., Poels, G., Rolland, C., Trujillo, J., Yu, E., Zimányie, E. (eds.) ER Workshops 2007. LNCS, vol. 4802, pp. 296–305. Springer, Heidelberg (2007)
17. Hall, J., Jackson, M., Laney, R., Nuseibeh, B., Rapanotti, L.: Relating software requirements and architectures using problem frames. In: IEEE Joint International Requirements Engineering Conference (2002)
18. Hill, T., Supakkul, S., Chung, L.: Confirming and Reconfirming Architectural Decisions on Scalability: A Goal-Driven Simulation Approach. In: Meersman, R., Herrero, P., Dillon, T. (eds.) OTM 2009 Workshops. LNCS, vol. 5872, pp. 327–336. Springer, Heidelberg (2009)
19. Horkoff, J., Yu, E.: Qualitative, Interactive, Backwards Analysis of *i** Models. Computer, 43–46 (2008)
20. Inverardi, P., Muccini, H., Pelliccione, P.: Checking consistency between architectural models using SPIN. In: Workshop From Software Requirements to Architectures (2001)
21. Jian, Y., Li, T., Liu, L., Yu, E.: Goal-Oriented Requirements Modelling for Running Systems. In: 1st International Workshop on Requirements at Run-Time (2010)

22. Pohl, K., Sikora, E.: The Co-Development of System Requirements and Functional Architecture. In: Conceptual Modeling in Information Systems Engineering, pp. 229–246 (2007)
23. Lapouchnian, A., Mylopoulos, J.: Modeling Domain Variability in Requirements Engineering with Contexts. In: Laender, A.H.F., Castano, S., Dayal, U., Casati, F., de Oliveira, J.P.M. (eds.) ER 2009. LNCS, vol. 5829, pp. 115–130. Springer, Heidelberg (2009)
24. Lucena, M., Castro, J., Silva, C., Alencar, F., Santos, E., Pimentel, J.: A Model Transformation Approach to Derive Architectural Models from Goal-Oriented Requirements Models. In: Meersman, R., Herrero, P., Dillon, T. (eds.) OTM 2009 Workshops. LNCS, vol. 5872, pp. 370–380. Springer, Heidelberg (2009)
25. Moody, D., Heymans, P., Matulevicius, R.: Visual syntax does matter: improving the cognitive effectiveness of the $i*$ visual notation. Requirements Engineering Journal 15(2), 141–175 (2010)
26. Morales, E., Franch, X., Martinez, A., Estrada, H.: Considering Technology Representation in Service-Oriented Business Models. In: 5th IEEE Workshop on Requirements Engineering for Services, Germany, pp. 482–487 (2011)
27. Nuseibeh, B.: Weaving the Software Development Process Between Requirements and Architectures. IEEE Computer 34(3), 115–117 (2001)
28. Oriol, M., Franch, X., Marco, J., Ameller, D.: Monitoring Adaptable SOA-Systems using SALMon. In: Workshop on Service Monitoring, Adaptation and Beyond, pp. 19–28 (2008)
29. Pahl, C., Gacitua-Decar, V., Wang, M., Bandara, K.Y.: A Coordination Space Architecture for Service Collaboration and Cooperation. In: Salinesi, C., Pastor, O. (eds.) CAiSE Workshops 2011, Part VI. LNBIP, vol. 83, pp. 366–377. Springer, Heidelberg (2011)
30. Pimentel, J., Franch, X., Castro, J.: Measuring Architectural Adaptability in $i*$ Models. In: 14th Ibero-American Conference on Software Engineering, CIBSE, April 27-29 (2011)
31. Pimentel, J., Lucena, M., Castro, J., Silva, C., Alencar, F., Santos, E.: Deriving Adaptable Software Architectures from Requirements Models: The STREAM-A approach. Requirements Engineering Journal (2011) (published online)
32. Pimentel, J., Santos, E., Castro, J.: Conditions for ignoring failures based on a requirements model. In: 22nd International Conference on Software Engineering and Knowledge Engineering, USA, pp. 48–53 (2010)
33. Pimentel, J., Santos, E., Castro, J.: Anticipating Requirements Changes – Using Futurology in Requirements Elicitation. International Journal of Information System Modeling and Design 3(2), 89–111 (2012)
34. Pinto, M., Fuentes, L., Troya, J.M.: DAOP-ADL: An Architecture Description Language for Dynamic Component and Aspect-Based Development. In: Pfenning, F., Macko, M. (eds.) GPCE 2003. LNCS, vol. 2830, pp. 118–137. Springer, Heidelberg (2003)
35. Qureshi, N., Perini, A., Ernst, N., Mylopoulos, J.: Towards a Continuous Requirements Engineering Framework for Self-Adaptive Systems. In: 1st RE @ Run-Time (2010)
36. Razavizadeh, A., Cîmpan, S., Verjus, H., Ducasse, S.: Software System Understanding via Architectural Views Extraction According to Multiple Viewpoints. In: Meersman, R., Herrero, P., Dillon, T. (eds.) OTM 2009 Workshops. LNCS, vol. 5872, pp. 433–442. Springer, Heidelberg (2009)
37. Shaw, M., Garlan, D.: Software Architecture: Perspectives on an Emerging Discipline. Prentice Hall (1996)
38. Taylor, R., Medvidovic, N., Dashofy, I.: Software Architecture: Foundations, Theory, and Practice. John Wiley & Sons (2009)
39. Yu, E., Giorgini, P., Maiden, N., Mylopoulos, J. (eds.): Social Modeling for Requirements Engineering. The MIT Press, Cambridge (2011)

Experience on Building an Architecture Level Adaptable System

Xu Zhang and Chung-Horng Lung

Department of Systems and Computer Engineering
Carleton University, Ottawa, Ontario, Canada
{zhangxu,chlung}@sce.carleton.ca

Abstract. Distributed and concurrent systems have become common in enterprises, and the complexity of these systems has increased dramatically. The self-adaptive feature can be advantageous for complex systems, because it can acclimate to a dynamically changing environment. To achieve this goal, this paper presents a Self-Adaptive Framework for Concurrency Architecture (SAFCA). SAFCA includes multiple concurrency architectural alternatives and is able to adapt to an appropriate architecture based on changes in the environment and the control policy. With an autonomic control, SAFCA can handle bursty workloads by invoking another architectural alternative at runtime instead of statically configured to accommodate the peak demands, which requires higher system resources even when they are not needed. Experimental results demonstrate that SAFCA can improve performance. The experience can be useful for building complicated systems that have multiple configurations or diverse demands, such as cloud computing.

Keywords: software adaptation, software architecture, software performance engineering, concurrency patterns.

1 Introduction

The average complexity of computer systems and the number of computing devices in use have been increasing dramatically [3]. As a result, IT personnel have to shoulder the burden of time-consuming (and therefore expensive) supporting tasks such as configuration, maintenance and system performance evaluation [5]. Further, manual control of a large distributed computing system is invariably prone to errors.

The goal of autonomic computing, initiated by IBM in 2001 [3], is to define rules for a system to control its behavior so that the system regulates its actions to automatically configure, heal, protect, and optimize itself [5]. More research efforts and IT companies have launched research projects related to autonomic computing recently [7] including in the area of distributed and concurrent applications, due to their high complexity.

Furthermore, the Internet is known for its dynamic nature. Under normal circumstances, the growing server farm is more than adequate to handle regular traffic demands. However, during special events, the server farm may become unavailable

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 171–179, 2012.

due to unprecedented demands [10]. On the other hand, many systems today are configured initially and statically to handle worst-case scenarios; in other words, systems are over-provisioned. The problem with this approach is that many resources will be wasted in normal condition scenarios.

Another challenge in software architecture is to conduct evaluation for the performance perspective, as there are usually uncertainties and lack of concrete data early in the life cycle. The problem could become more difficult if multiple software architectural alternatives exist.

This paper presents a self-adaptive system, self-adaptive framework for concurrency architectures (SAFCA), that supports software adaptation at the architecture level for the distributed and concurrent problem domain. The self-adaptive system consists of multiple software architectural alternatives. The system can adapt to changing demands at runtime by using the appropriate alternative. The main objective of the adaptation is to better utilize system resources and increase performance.

This paper is organized as follows: Section 2 provides the background information on distributed and concurrency architectures. Section 3 describes the self-adaptive framework, SAFCA. Section 4 presents experiments and the performance results. Finally, Section 5 is the conclusion.

2 Background

This section provides brief background information on the primary concurrency architectural patterns used in the paper.

2.1 Concurrency Architectural Patterns

Distributed and concurrent programming has been widely used in many applications. There are several known concurrency architectural alternatives: single-thread, dynamic thread Creation (DTC), Half-Sync/Half-Async (HS/HA), and Leader/ Followers (LFs) [8]. Both HS/HA and LFs are thread pool-based approaches. According to our previous performance modeling studies using the layered queuing networks [11], HS/HA has higher efficiency than LFs for high demands. Therefore, HS/HA is chosen as the main thread pool-based architecture used in the SAFCA.

DTC, on the other hand, is selected for the study, because it is the simplest and the common design that does not have a thread pool. Our framework actually supports LFs as well. But the LFs pattern is not included in this comparative evaluation.

Dynamic-thread-creation (DTC): The DTC architecture is commonly used in multi-threaded programming, especially for server applications. DTC is based on the idea of a thread-per-request. DTC creates a thread for each new request. A thread dies after processing its request [9]. DTC is relatively easy to program. However, a thread has to be dynamically created per request, there is an overhead associated with it.

Half-Sync/Half-Async (HS/HA): In order to simplify programming without unduly reducing performance, HS/HA decouples asynchronous and synchronous service processing in concurrent systems [8].

External input first arrives at the asynchronous layer. After the asynchronous layer processes the input, the request is stored in a queue in the queuing layer. The function of the queuing layer is to buffer input from the asynchronous layer to the synchronous layer and inform the synchronous layer that input is now available. Worker threads in the synchronous layer retrieve input from the queue and process the input further. Layers are independent from each other and can perform operations concurrently.

2.2 Self-adaptive Systems

The aim of self-adaptive or autonomic computing is to provide a solution that runs services with minimum or lower cost, capable of both scaling up and scaling down the system resources responding to dynamic demands. This requires system elasticity, in terms of allocating or using resources as they are needed. Therefore, systems must adapt to changes in the environment quickly and since manual server reconfiguration has been shown to be inadequate [4], self-adaptive solutions are better suited for such problems. Autonomic computing has received more attention recently in software engineering, e.g., [1][2][6], mainly due to the increased complexity of systems. One crucial research issue is to build the capability into the system to adapt its behavior in response to the dynamically changing environment.

In [12], we also demonstrated the feasibility of SAFCA using a simple approach based on queue length. The idea of the queue length-based approach is simple, i.e., if the threshold is passed, adaptation will be invoked. But it is challenging to determine queue length and oscillations may happen. In this paper, we devise a policy to detect the traffic burst which triggers software adaptation.

3 Self-adaptive Framework for Concurrency Architectures

SAFCA is designed to support adaptation at the architecture level during runtime based. In other words, adaptation occurs from one architectural alternative to another. This section first describes the main concept of the approach. The framework and the main components are then presented. After that, the self-adaptation scheme is described. The monitoring mechanism of busty traffic and the policy of managing busty traffic autonomically will be demonstrated.

3.1 Problem Scope and Main Concept

The paper considers three conditions: 1) when an overloading request burst occurs, the framework can achieve acceptable response time and decrease the message or packet loss ratio; 2) the software resource (for example, the number of threads) utilization under normal load condition can be minimized; 3) the self-adaptive framework must be practical in the sense that it can be easily implemented.

HS/HA works well under normal workload conditions. However, if a burst occurs, the pre-configured size of the thread pool becomes the performance bottleneck for HS/HA. On the other hand, DTC can create a large number of threads to handle the sudden burst of requests. But the overhead of thread creation and destruction in DTC makes it inferior, compared to HS/HA, under normal workload condition. For this reason, the self-adaptive system containing both HS/HA and DTC architectures is built to validate the concept of self-adaptation. In normal conditions, HS/HA is used. If a burst arrives (based on the burst detection policy), DTC will be activated immediately, because HS/HA has a pre-configured pool size. During this period, both HS/HA and DTC alternatives are running concurrently or simultaneously (for a multi-core system). The approach is compared against a system running either HS/HA or DTC alone for performance evaluation.

3.2 Overview of the Framework

In order to support the tasks described in Section 3.1, Figure 1 depicts the framework and the four main components. The *Monitor* component gathers information regarding the queue length, response time, arrival rate, and number of threads currently running (number of threads that have received a request but has not sent a reply). The *Decider* then computes statistical-average-quantities, such as average queue length, average response time, etc. Based on collected and calculated information, the *Decider* decides if any action should be taken. For example, if the queue length exceeds a threshold, the *Decider* notifies the *Executor* component that the current workload condition is heavy. The *Executor* will then instruct the *Architecture Manager* to put new requests into the appropriate queue.

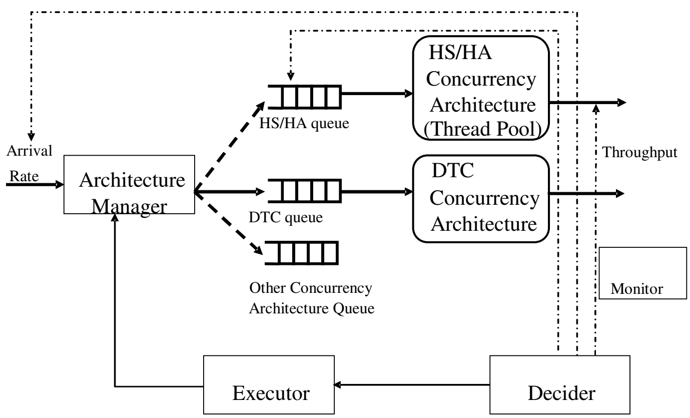


Fig. 1. Self-Adaptive Framework for Concurrency Architectures Overview

The framework consists of multiple architectural alternatives. Currently, three alternatives are included: DTC, HS/HA, and LFs (not shown in Figure 1 and not included in performance evaluation). Each architecture alternative has its own queue.

The *Architecture Manager* has a high scheduling priority so it can respond to incoming requests immediately. Once the destination queue becomes full, the *Architecture Manager* drops any new requests.

3.3 Self-Adaptation Policy for Bursty Traffic

The self-adaptation policy is designed so that HS/HA is the default architecture. During normal scenarios, DTC is not active and the *Architecture Manager* puts new requests in the *HS/HA queue*. The trigger to self-adaptation is based on arrival rate and average response time, see Section 3.4.

When an arrival burst occurs and the response time also increases over a certain level, the *Architecture Manager* begins to place new requests to *DTC queue* to deal with bursy demands. When the burst is over and the arrival rate returns to the pre-burst level, the *Architecture Manager* puts new requests to the original *HS/HA queue*.

3.4 Burst Detection Policy

Monitoring and measurements are two important elements for autonomic computing. Monitoring and measurements are used for burst detection in our approach. Detection of a burst could range from a simple threshold-based approach according to the queue length [12] to a sophisticated method. This paper presents an approach based on both the arrival rate and the average response time for message processing. Arrival rate alone cannot indicate if the system has reached its capacity. Therefore, the average response time is also used.

In the burst detection policy, the standard deviation (σ) of previous arrival rates and the mean arrivals in a pre-configured interval are used.

Assume *r1, r2, r3,..., rn* are arrival rates for sampling intervals *1, 2, 3,..., n,* respectively. If the difference between the current arrival rate and the mean arrival rate is greater than the standard deviation of previous arrival rates (σ), then a burst is assumed to have occurred. However, this does not mean that the system is overloaded, but a self-adaptive action needs to be taken. If the difference between the current response time and previous response time also increases by *y%* (a pre-configurable parameter) of the previous response time, then the *Decider* notifies the *Executor* to take actions and the mean arrival rate is reset to zero. From this point on, a new mean arrival rate and a new standard deviation will be calculated.

If the current arrival rate is less than the mean arrival rate, and their difference is greater than the standard deviation of the arrival rates (σ), then the burst is assumed to be over. The *Decider* notifies the *Executor* to free resources and mean arrival rate is reset to zero. From this point on, a new mean arrival rate and a new standard deviation will be calculated.

4 Experiments and Analysis

This section presents experiments conducted and results. The performance of SAFCA is compared with that of standalone HS/HA and DTC without adaptive control.

4.1 Experiment Settings

The experiments consider a multi-tier system. The server receives message or requests from multiple clients, and the traffic generated by those clients contains random bursts. Each request received by the server is processed including CPU-bound operations and I/O-bound operations, and then a reply is sent back to the client.

Our test bed consists of one server machine (3.0 GHz Pentium 4 systems with 3.49 GB of RAM) and a client machine (3.0 GHz Pentium 4 systems with 3.49 GB of RAM) connected to a Phoebe Ethernet Switch (8-Port 10/100Mbps Auto/MDIX). SAFCA is developed with SUN JDK 1.6 as the Java platform running on Microsoft Windows XP Professional on the server side. Multiple clients generate traffic and send requests to the server. The client traffic generator is also developed with the same platform as that of the server. A number of experiments have been conducted. TABLE I lists the parameters used for the experiments.

Table 1. Parameters used for experiments

Experiment Parameters	Value
burstAverageArrivalRate	200 messages/sec
normalAverageArrivalRate	50 messages/sec
maxBurstDuration	20 internvals
minBurstDuration	10 intervals
maxNormalDuration	30 intervals
minNormalDuration	20 intervals
sampleIntervalLength	5 sec
queueSize (HS/HA)	25
queueSize (DTC)	25
threadPoolSize	60

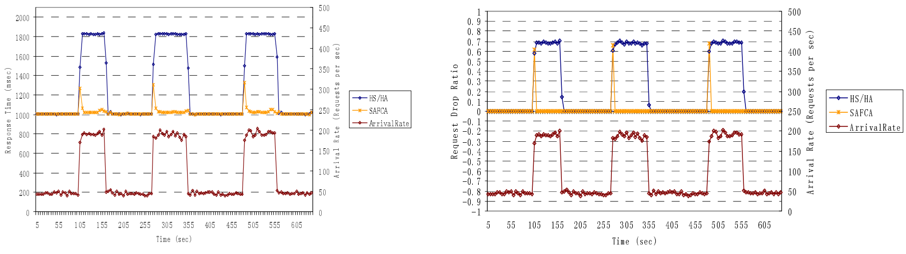
4.2 Performance Evaluation of SAFCA

This section evaluates the performance of SAFCA with HS/HA and DTC in terms of response time, request drop ratio, and utilization. As described in Section 3.3, the policy of SAFCA is to initially send requests to the HS/HA queue. When the arrival rate has increased by more than one standard deviation from the previous average arrival rate and the response time also has increased by more than 20% (a configurable value), SAFCA sends new requests to the DTC queue. If the arrival rate has decreased by more than one standard deviation, SAFCA sends new requests back to the HS/HA queue. The results show that SAFCA offers better performance.

SAFCA and Standalone HS/HA

Figure 2(a) depicts the response times for SAFCA and standalone HS/HA on the primary y-axis (on the left) and the arrival rate on the secondary y-axis (on the right). The response time of SAFCA is low during normal workload because it uses HS/HA which is more efficient for normal workload. During bursts, the response time is still low because SAFCA dynamically invokes DTC to cope with the bursts. This arrangement is efficient because with HS/HA, the configured number of threads (60) is sufficient during non-burst periods and no new threads are created. On the other hand, DTC can accommodate high demands during burst periods because it can create more threads. For standalone HS/HA, the throughput bottleneck is due to its fixed number of threads.

The spikes of the response time are primarily due to bursts and the monitoring interval. If the interval is reduced, the spikes can be shortened.



(a) (b)

Fig. 2. Comparison of Response Time (a) and Drop Ratios (b) for SAFCA and HS/HA

Figure 2(b) illustrates the request drop ratio of for SAFCA and HS/HA. The drop ratio of SAFCA is close to 0 (except when a burst first starts) during both normal workload and burst workload. Again, the drop ratio of SAFCA for the initial burst periods could be reduced by shortening the monitoring or sampling interval.

Figure 3 compares the thread utilization for SAFCA and standalone HS/HA. The thread utilization is always low (between 10% and 20%) for HS/HA because the thread pool size is fixed. However, because SAFCA uses DTC to create more threads when needed during bursts, the resource is better utilized with utilization mostly in the range of 60% to 70%.

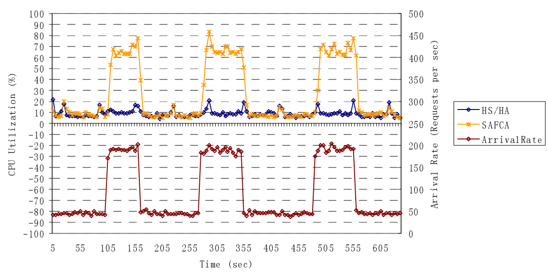


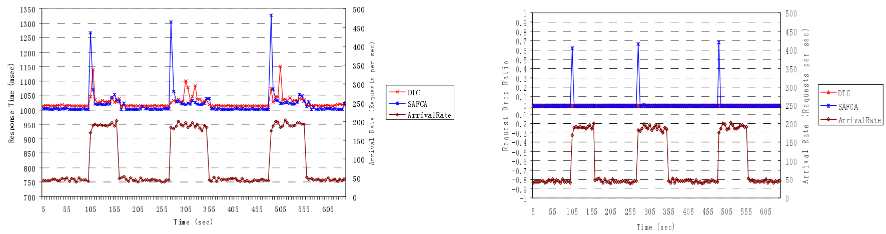
Fig. 3. Comparison of CPU Utilization for SAFCA and HS/HA

SAFCA and Standalone DTC

In this experiment, the response time, the request drop ratio, the thread utilization, and the number of created threads are measured for performance evaluation. The results show that SAFAC has better performance in most cases.

Figure 4(a) shows the response times for SAFCA and DTC. Except when the burst first started, SAFCA has a better response time than that of DTC in most cases. Again, the sharp spikes of SAFCA can be mitigated using shorter monitoring length.

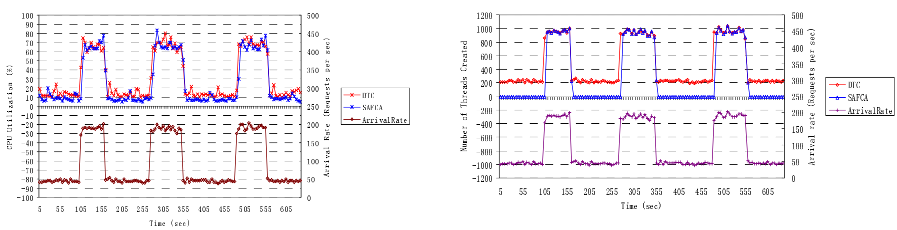
Figure 4(b) presents the drop ratio of for SAFCA and DTC. Both SAFCA (except when the burst first starts) and DTC have a loss ratio close to 0.



(a) (b)

Fig. 4. Comparison of Response Time (a) and Drop Ratios (b) for SAFCA and DTC

Figure 5(a) illustrates that SAFCA has similar thread utilization as that of DTC in both normal workload and burst workload conditions.In terms of resource usage, Figure 5(b) shows that DTC creates more threads and consumes more resources than SAFCA during normal workload condition. Each thread created requires memory space, CPU cycles, and the operating system overhead for thread creation/destruction. As depicted in Figure 5(b) in one sampling interval, DTC creates about 200 new threads even under normal conditions. In comparison, no additional threads are created dyanmically or only 60 existing threads in the thread pool of SAFCA can handle the normal workload. Since normal workload periods are typically much longer than burst periods, SAFCA is more resource efficient than DTC.



(a) (b)

Fig. 5. Comparison of CPU Utilization (a) and Number of Threads Created (b) for SAFCA and DTC

5 Conclusions

In order to effectively utilize resources under dynamic workloads, a self-adaptive framework, SAFCA, was proposed and developed. According to the results obtained from a number of experiments, SAFCA improved the performance under various workloads through an adaptive mechanism. In comparison to the standalone HS/HA and DTC, SAFCA exhibited performance gains without the need of over-provisioning as often adopted for thread pool-based approach. Under normal workload conditions, SAFCA has a better resource usage than DTC-only system. The concept could be useful for other applications to support scaling up and down of a system or cloud computing where many configurations and diverse resources and demands exist.

References

[1] Brun, Y., Di Marzo Serugendo, G., Gacek, C., Giese, H., Kienle, H., Litoiu, M., Müller, H., Pezzè, M., Shaw, M.: 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)

[2] International Symp. on Software Engineering for Adaptive and Self-Managing Systems (2006-2012)

[3] Enterprise Management Associates, Practical Autonomic Computing: Roadmap to Self-Managing Technology. Tech. Rep., IBM, Boulder, CO (January 2006)

[4] IBM Autonomic Computing Architecture Team, An Architectural Blueprint for Autonomic Computing, Tech. Rep., IBM, Hawthorne, NY (June 2006)

[5] Kephart, J., Chess, D.: The Vision of Autonomic Computing. Computer 36(1), 41–52 (2003)

[6] Kramer, J., Magee, J.: Self-Managed Systems: An Architectural Challenge. In: Proc. of the Future of Software Engineering, FOSE, pp. 259–268 (2007)

[7] Muller, H.A., et al.: Autonomic Computing, Tech. Rep., Software Engineering Institute (April 2006)

[8] Schmidt, D., Stal, M., Rohnert, H., Buschmann, F.: Pattern-Oriented Software Architecture, Patterns for Concurrent and Networked Objects. Wiley (2000)

[9] Welsh, M., et al.: A Design Framework for Highly Concurrent Systems, T.R., UC Berkeley (2000)

[10] Welsh, M., Culler, D.: Adaptive Overload Control for Busy Internet Servers. In: Proc. of the 4th USENIX Conference on Internet Technologies and Systems, p. 4 (March 2003)

[11] Zhang, X., Lung, C.-H., Franks, G.: Towards Architecture-based Autonomic Software Performance Engineering. In: Proc. of the 4th Conference on Software Architectures, CAL (2010)

[12] Zhang, X., Lung, C.-H.: Improving Software Performance and Reliability with an Architecture-Based Self-Adaptive Framework. In: Proc. of IEEE COMPSAC (2011)

The 4x6 Tiered Architecture Method: An Approach to the Design of Enterprise Solutions

Ethan Hadar[1], Irit Hadar[2], Gabriel M. Silberman[1], and John J. Harrison Jr.[1]

[1] CTO Office
CA Technologies
{ethan.hadar,gabriel.silberman,jay.harrison}@ca.com
[2] Department of Information Systems
University of Haifa
{hadari}@is.haifa.ac.il

Abstract. Enterprise architecture software design is all about composing applications to assemble value-added solutions rather than standalone products. Yet, each product and technology may have been designed and developed separately because of software engineering practices, management control over the deliverables, or technology acquisitions. To promote efficient assembly, solutions must be architected in a similar style, adhering to fundamental design principles while leveraging capabilities available in modern environments and relevant platforms. Furthermore, business agility and cost requirements dictate the identification of common capabilities and their development as reusable components across products and solutions. The 4x6 Tiered Architecture Method presented in this paper imposes a structured design, in terms of steps to follow, structure and documentation, for the logical view of an enterprise solution. Application of the 4x6 method to the analysis of an enterprise solution yields a six-tiered architecture structure and an abstract architecture specification. This specification expresses the various components, dependencies and design patterns using a graph-based data model (or “architecture catalog”) and blueprint, the latter expressed as both a diagram and XML document. The 4x6 Method has been applied in practice; this experience indicates that this method results in higher quality architecture and requires lower effort for both constructing and reviewing the architecture and its documentation.

Keywords: Design Tools and Techniques; Software Architectures; Domain-specific architectures; Patterns.

1 Introduction

The architecture design for an enterprise solution involves a number of challenging decisions, including the expected transaction load, response times, volume of users, number of integrated systems, options to deploy as a stand-alone, on-premise, or as a Software-as-a-Service (SaaS) solution. To address these issues and produce a good design, architects employ design patterns [1] and, in many instances, reuse existing components and integrated services to solve known problems with known solutions

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 180–191, 2012.

[2]. The design intent is to invest the most (new) effort while reusing existing assets as much as possible. To further enable the combination of existing solutions, and their replacement as future technologies emerge, requires us to characterization them in terms of overall capabilities and non-functional characteristics from the customer perspective [3].

In short, as long as the product or solution is not an isolated instance, most design activities will deal with adding capabilities to existing modules, integrations with external technologies and services, and/or refactoring and evolution of the architecture structure. The challenge for the solution architect is to provide a quality design for separate structures resulting in easy to understand, out-of-the-box components, with a set of (estimated) characteristics for the combined result.

To tackle these challenges, we developed a model-driven architecture method, called the *CA Four Architecture (C4A)*, an extension of the C3A approach described in [4]. As in other Model Driven Architecture (MDA) approaches, multiple views over a single model enable the capture of design intent and multi-dimensional module characteristics, while documenting architecture decisions. The proposed methodology uses four diagrams, representing four views, iteratively developed and refined through a series of analysis, design and delivery steps, as shown in Figure 1.

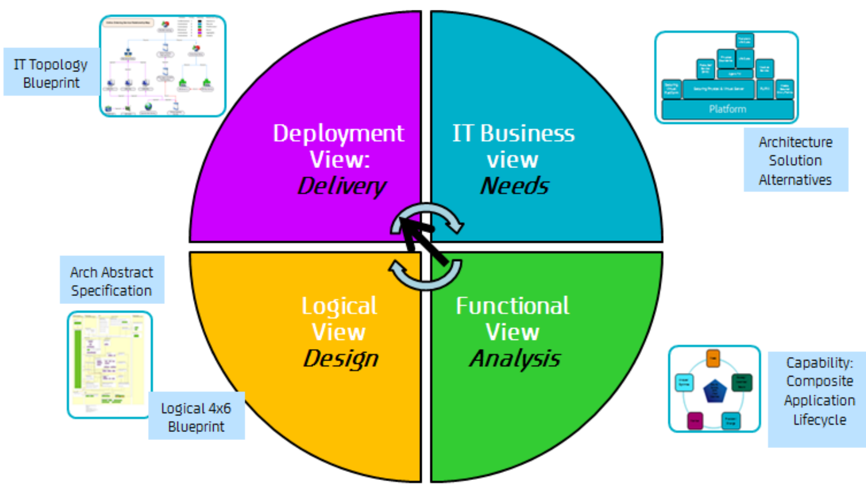


Fig. 1. C4A Architecture Views and Process

Although similar to the 4+1 view model by Kruchten [1], C4A employs fewer diagrams than the eight suggested by Kruchten and addresses a larger portion of the full software development life cycle, including a target reference architecture and evolution plan. C4A also integrates the design process via systematic analysis and design flow, which in the 4+1 approach is managed through integration with the Rational Unified Process [1].

As depicted in Figure 1, the C4A views include 1) an *IT Business View*, focused on the business rationale and "go-to-market" needs; 2) a *Functional View* to capture the "jobs-to-be-done" (JTBD) [5] capabilities using the customer's own business language and taxonomy; 3) a *Logical View* of the architecture software components as the entry point for a detailed design for the R&D team; and 4) a *Deployment View* to support issues such as configuration and deployment of components using on-premise hosting, virtualization and /or leveraging of external (e.g., Public Cloud) environments.

The integrated analysis and design process, introduced briefly in the following section, defines the architecture activities and artifacts, regardless of the software development lifecycle method (e.g., Agile, Waterfall or Incremental) it is intended to support. Its detailed description is beyond the scope of this paper. The remainder of this paper focuses on the Logical View as the fundamental pillar of the design phase, its design goals, and introduces the *4x6 Tiered Architecture Method* (or *4x6 Method,* for short) approach to logically architecting an enterprise solution.

Application of the 4x6 Method to an enterprise solution imposes a structured design, in terms of steps to follow, structure and documentation, and yields a six-tiered architecture structure and an abstract architecture specification. This specification expresses the various components, dependencies and design patterns using a graph-based data model (or "architecture catalog") and blueprint, the latter expressed as both a diagram and XML document. This structured approach enables a software product line assembly [6][7] of reusable patterns and components, with increasing development efficiency over time, and applicable to a whole domain, solution, sub-system, product, or a single component.

2 Analysis and Design Process

The C4A methodology is an attempt, based on best practices, to address the five architectural concerns presented in [6][7], namely economy, visibility, spacing, symmetry, and emergence.

The *Economy* design goals are to define and detect usable, unique IT modules and services and evolve toward value-added solutions built on top of common technologies and services. The *Visibility* design goals are to employ a unified service and data language across the various IT domains, and to implement systematic architecture taxonomy, symbolic representations and patterns across the solution's various levels of documentation. The *Spacing* design goals are to provide replaceable modules which are loosely-coupled and based on a services model as well as produce granular pre-packed smaller offerings of dedicated (composite) services yielding reusable commodities. Finally, The *symmetric* main design goal is to support a service abstraction and orientation approach, such as the virtual IT services found in a Software-as-a-Service (SaaS) delivery modality, regardless of the actual delivery mode. Such a decoupling between a service and its actual implementation enables symmetric provisioning and consumption of the service, thus supporting the construction of a composite application in a supply-chain manner [8]. Additional goals are to standardize over time by using common technology and IT services to embed within the IT integration framework. The *Emergence* design goals are to detect and adjust to changes in underlying commodities, and explore new ways of interactions among systems and users (e.g. via mobile devices, virtual appliances, etc.).

To address the architectural concerns listed above, the analysis and design process embedded in the C4A methodology includes two analysis and two design phases, to correspond with the four views shown in Figure 1. These are briefly outlined below.

The *IT Business Analysis Phase* operates within the IT business view of the subject IT solution. During this phase the solution architect creates a business value proposition according to high-level business expectations. Various graphic means may be used to represent the result of this phase, including an architecture stack, flow or PERT charts, and the like. For our example, shown in Figure 2, we chose a stack representation. In our “hello world” solution, its value proposition consists of mobile interaction in a multi-lingual, user-role-sensitive scenario. During the business analysis phase we separate our offerings into two basic blocks, Sentence Building and Multilingual Adjustment. The former composes a structure of English strings, while the latter has replacement capabilities of these strings with corresponding sentences in other languages. Only after successfully building these two blocks, one can address the need to limit access to the functionally by the user, depending on their identity and corresponding role(s) within the enterprise. Once this intermediate block is properly addressed, attention can shift to the question of where the solution is accessed from, a desktop or mobile device. Thus, the full solution features a gradual implementation roadmap for continuously measured progress.

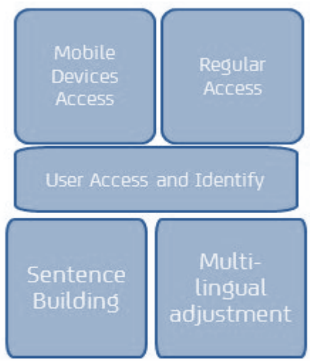


Fig. 2. Stack representing a mobile interaction in a multi-lingual, user-role-sensitive scenario

The *Functional Analysis Phase* of the process works inside the functional view. This analysis captures the solution’s capabilities organized as an IT service lifecycle. These capabilities are structured as jobs-to-be-done (JTBDs) defining the action verb, the object of the action, and the contextual classifier [5], driving technological integrations and ultimately defining the customer architecture assimilation roadmap(s). Following our example, the capabilities of the multilingual adjustment offering will be: (JTBD1) replacing English sentences with the corresponding text in a different language, and (JTBD2) on demand provisioning of a number of language libraries. Each JTBD is tagged and capabilities are collected according to these tags, to be later mapped to an implementation component in the logical design view. In our example, JTBD1 is tagged as "change," and JTBD2 as "model". Although any tagging is possible, in C4A our IT service lifecycle phases/tags are model, assemble, change, monitor, and optimize.

The *Logical Design Phase* corresponds to the design view and focuses on integration (via APIs) and the functional layer (GUI, if applicable), mapping structures to the components in the functional capability view created by the previous phase. Further, the architect examines possible mappings of available design patterns across the logical layers, and plans alternative roadmaps for evolving the architecture. Understanding the logical and physical dependencies among the various components is critical to correctly estimate the overall quality attributes and performance characteristics of the solution being built. For example, a component depending on a lower reliability module needs to account for that exposure when its overall score is calculated. The detailed application of this phase to our "hello world" solution is provided as part of the detailed discussion in the next section.

The *Deployment Design Phase* acts within the design view and its objective is to gather the various components making up a particular instance of a solution. The main activity in this phase is the definition of computing resource requirements and constraints on their nature (physical, virtual, or Cloud) to fulfill the needs of the logical components. In the context of our "hello world" sample solution, the design could prescribe the multi-language components to be consumed as a service and thus would not require deployment. Also, the design could require the mobile device component to be highly scalable and use cache memory for improved performance, while the sentence building component needs to be restricted to a maximum memory footprint, yet can be run anywhere as a stateless server.

The overall design intent of a logical architecture and its blueprinting implementation must:

- enable managers to leverage resources across their portfolio [6][7][9];
- separate non-unique technologies into interchangeable consumable commodities [6][7];
- enable the composition of a technology from underlying patterns [4];
- provide a modular architecture and appropriate evolution roadmap [10];
- increase the overall quality attributes of the architecture structure [11][12][13]; and
- enable the structured estimation of aggregated system quality attributes [8].

The challenge is thus to create an architecture focused on consolidation and optimization of component and service (re-)use, while increasing the overall solution's quality attributes. We must do so while bundling all of the design requirements, constraints, principles and directives into a single architecture blueprint reflecting a good enterprise solution design.

In the following we examine the 4x6 Logical View and its capacity to address the above challenge, by enabling an architect to systematically model and superimpose existing and future technologies, and design the architecture evolution of an individual product or enterprise solution.

3 The 4x6 Logical View

The structure of the architecture obtained by applying the 4x6 Method consists of 4 conceptual tiers, or stereotypes, that underlie 6 logical tiers, yielding the “4x6” designation. The four conceptual tiers (the “4” in 4x6), are defined in the C3A approach [4], namely: 1) business integration; 2) functional architecture; 3) system architecture; and 4) cross-concerns. As for the six logical tiers (the “6” in 4x6), they now include a mapping of the classical three-tier architecture pattern (Presentation, Business, and Storage), plus three additional tiers. The resulting tiers are: 1) virtual IT services (a business integration stereotype); 2) views (corresponding to the classical Presentation tier, a functional architecture stereotype); 3) business logic (corresponding to Business, a system architecture stereotype); 4) data (corresponding to Storage, and also a system architecture stereotype); 5) integrated services (another business integration stereotype); and 6) common components (a cross-concerns stereotype). The 4x6 logical view layout is depicted in Figure 3.

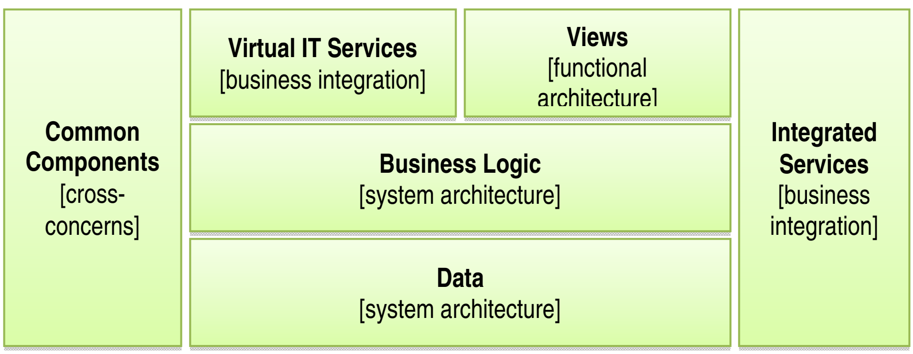


Fig. 3. The 4x6 logical view layout; square brackets indicate the kind of conceptual tier, or stereotype

It is worth noting C4A uses the four stereotypes (business integration, functional architecture, system architecture, and cross-concerns) in its 4x6 view in order to cater to a different stakeholder, namely external integrators, functional architects, system architects, and common components managers, respectively.

A hypothetical layout for our “hello world” solution is depicted in Figure 4. In it we see the four stereotypes (outer rectangles, colored yellow in the modeling tool) and three abstraction levels. Level 0 (middle rectangles, colored blue in the tool) represents high-level modules or sub-systems. Below it, Level 1 (inner rectangles, colored green) contains deployable components, which may be removed and replaced with similar components without affecting the rest of the system or requiring the replacement of a full Level 0 module. Level 2 (colored orange, not used in this example) contains an internally cohesive set of components that is usually deployed or managed as a unit.

Color is also used to provide the state of components, either existing or future, in a single view. Those components intended for future development (or modification) are left white by default (e.g., support for right-to-left languages in Figure 4) and can be set by the architect to any color in order to reflect the timing (or extent) of the implementation.

A summary description explicitly calling out the scope of new product release(s) within the overall solution, as well as their long-term architecture roadmap, are also produced during the building of the solution's layered architecture. In our "hello world" example (Figure 4), the blueprint suggests that most of the components existed and were implemented, due to their color-coding as either blue (Level 0) or green (Level 1). The additional capability being added (in white), to translate languages read/written from right to left, is limited to string building. The purple color (rectangle *Language format* in this example) is used to signal the timing (e.g., next release) for the additional capability to appear in the solution. These new components are owned by the development team, since they reside at the middle business tier.

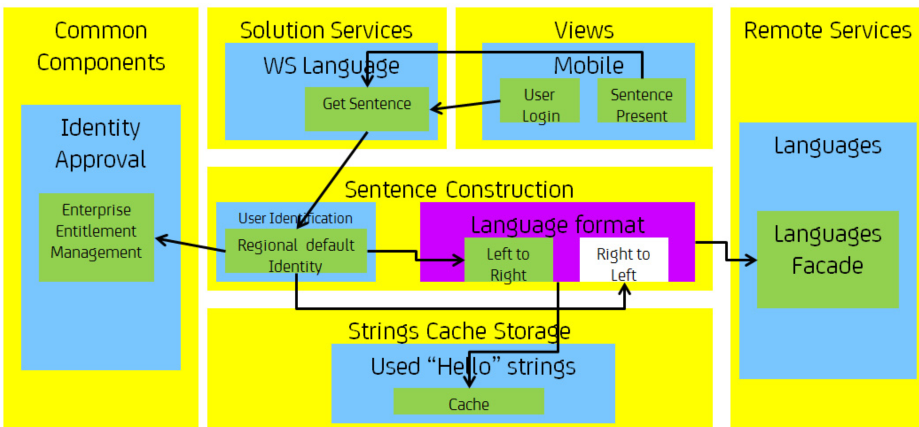


Fig. 4. The 4x6 layout for the "hello world" solution

The blueprint also shows (in the External Services tier) the usage of external language sources, provided via a façade design pattern. On the other hand, the association of a geographical region with a user and the corresponding selection of a default language are done through querying a common component of the enterprise. By definition, common components are not part of the architecture development, but are instead treated as if they came from a third party, but with a known implementation. Thus the placement of components among the six tiers and their color-coding enable the architect to provide implicit information about the characteristics of a solution to other designers and observers. Naturally, the simple example presented here may evolve into many components, depending on the granularity and complexity of the solution. Yet, they all adhere to the same design principles presented in this paper.

This logical blueprint, which contains components properties and attributes, may also be used to create a top level (abstract) design specification document for a product or solution planned release, as well as documentation for the product/solution family as a whole. Still, the blueprint's most valuable use is in assisting with the definition of architecture evolution roadmaps, from an existing state to new releases over a given period of time.

In the next sub-sections we explore ways for an architect to most effectively use the 4x6 logical view to convey architectural intent, followed by an examination of the six logical layers. We wrap up this section with a brief description of a tool we developed to assist the architect with the creation of the 4x6 logical view.

The 4x6 logical view captures the software elements of a solution, including components and their logical interactions. We found architects benefit by considering four design value propositions as they create the 4x6 logical view. These are *components coupling* for obtaining loosely-coupled structures for future assistance with system development and deployment, *components cohesion* for separating the solution's intellectual property from needed functionally, *meaningful architecture naming* for conveying the business value of the design elements, and *composition of aggregated tiers* for dealing with solutions composed of a single technology.

When constructing an architecture using the 4x6 Method, the loose coupling between its tiers hints at their potential separate deployment. The distribution onto six tiers is also meant to reflect logical constrains or concerns for the system design, captured as six logical layers. Logical layers do not impose coupling restriction, but rather a division of responsibility, defining a weak cohesion relation or grouping of components based on type, not cohesiveness based on the same functionality. These are important for any design, but more so for existing solutions and products not featuring six logical layers built as six separate tiers. These solutions have the components to fulfill the intent of the logical structures, and will gradually refactor their structure into corresponding tiers, enabling rapid integration with other products to form a larger solution.

Our aim is to work with or towards tiers, but most current product and solution designs feature only layers. Therefore, in the reminder of this section we shall use the term "layer" to replace "tier," keeping in mind the logical rather than the physical intent, as explained above.

We start with the *Views Layer*, which addresses the visibility, spacing and emergent concerns of an architecture. This layer contains the user interface(s) and presentation rendering adjustments, which are involved in human interaction. The rendering adjustments adapt the interface to a particular device type or presentation technology, such as mobile, tablet or desktop, browser, Linux or Windows, flash or AJAX, etc. Therefore, this view supports the "View" in the Model-View-Controller design pattern [6][7], whereas "Model" and "Controller" are part of the Business Logic Layer (see below). User interface and human factors design are part of this layer's best practices. As a result, the three main Level 0 components in this layer should be "Multi-device UI façade," "Reports Publisher," and "Content Synchronization controller."

Next is the *Virtual IT Services Layer*, to address all five architectural concerns, namely economy, spacing, emergent, visibility, and symmetry. The main purpose of this layer is to encapsulate programming namespaces, and in many cases it merely holds the system software development kit (SDK) in several interoperability formats to support the external activation of business transactions. Consequently, its two main Level 0 components are a regular SDK, called "Published Service," and one employing a domain unified language, called "Published Canonical Data."

The *Business Logic Layer*, addresses spacing, economy, symmetry and visibility concerns. This layer encapsulates the intellectual property of the offered technology. Any interaction with other layers should be surrounded by boundary components, model interfaces, or the data layer. The boundary components maintain a clean design-to-test approach, consumed via the "Internal Domains Specific Model" Level 0 component, and the Level 1 "Web Interfaces" component. The third component is the "Data Access Layer" or DAL. The DAL abstracts the persistency technology and exposes the CRUD operations (Create, Read, Update, Delete) for translating data into the object format understood by the Business Logic Layer. An example of such a translation is needed for stream-based non-persistent data retrieved from agents and monitors

The *Data Layer* addresses the concerns of visibility, spacing and emergent. Its focus is on increasing data resiliency, improving I/O performance, and providing ease of content scaling and expansion.

Focusing our attention on the bracketing layers in Figure 3, we first look at the *Integrated Services Layer*, which addresses the economy, emergent and symmetry concerns. This layer abstracts remote activation of (Web-) services, presumably commoditized ones, and presents a single gateway to general services such as reporting, logging, identity management, message bus, and more. In a cloud-based era, these services may be consumed from a remote vendor, and paid on a per-use basis. By encapsulating remote activations and delegating local calls, directly or indirectly, this layer supports future advances in technologies instead of rigid deployment and coupling. Therefore, this layer deals with how to design issues such as a good API, Generic versus Specific API, Web Services Variation Façade, Canonical Data Model, Hub-and-Spoke, Broker, Observer, and Publish/Subscribe patterns.

The second bracket in Figure 3 is the *Common Components Layer*. This layer addresses economic design goals, containing embedded and deployed components that are considered part of the compound solution, although not constructed by the development team. The first option for using a common component is to enforce its installation regardless of the number of instances already deployed at the customer's site. The second is to conditionally install components only when they are not already present in the deployment environment. Notice that Common Components are part of the provider solution and must not be replaceable by the customer.

Recall that the 4x6 components proposed within each layer should be given archetypical names, hinting at the design patterns used to define its technological purpose. When the design is instantiated as a real blueprint, the business (value) name is added giving the component a (composite) meaningful name.

A modeling Tool: The 4x6 logical blueprint is implemented in a modeling tool called *CAM Logical Architecture* (CAM stands for CA Architecture Management) displayed in Figure 5. Each of the model's components has the following set of properties: interfaces, functional description, non-functional level of enterprise compliance (the so-called "ities," such as reliability, scalability, security, etc.), and resource requirements. The component blueprint, component properties and dependency information, and overall blueprint properties are stored in a component catalog. This catalog is used to produce reports and can translate content to other formats outside of the tool's technology. Also provided is a centralized library of captured information at the individual component level and blueprints, enabling information sharing and design reuse among CA's architects community.

The component catalog can effectively produce an Architecture Abstract Specification [11], automatically providing about 70% of the information needed for examination by the internal CA architecture review board. The modeling tool itself, which operates in either centralized or local mode, is based on the Eclipse Modeling Framework and the ECORE model as a standalone logical editor.

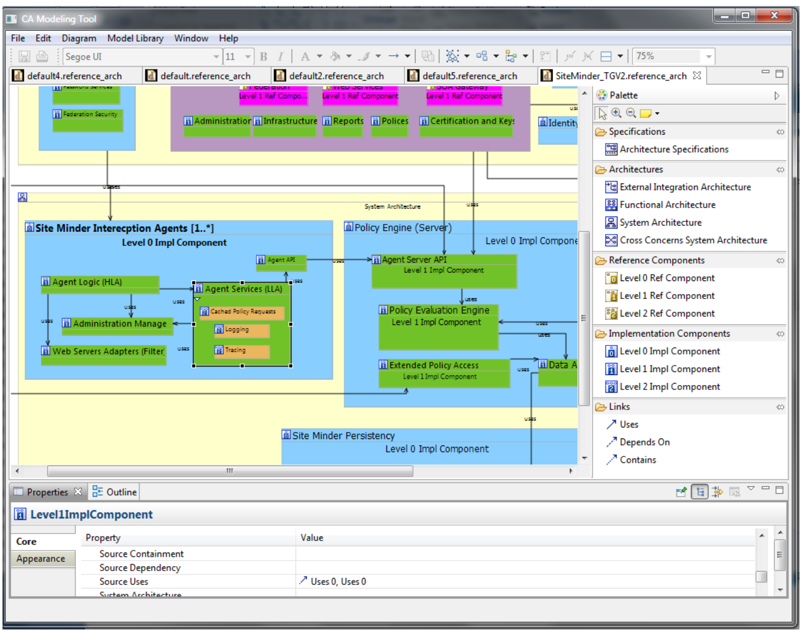


Fig. 5. The CAM Modeling tool for the 4x6 logical architecture

4 Applying the 4x6 Method in Practice

On January 2011 the 4x6 Method and CAM tool were presented at the CA corporate architects' conference in front of more than 300 architects, as part of a new method of modeling and capturing architecture information in an agile manner. It was enthusiastically received by many of the architects, who requested guidance to use the method and tool. Following the conference, educational material was recorded on video and made available as part of the delivery process.

Training on the tool and 4x6 approach takes less than an hour, assuming basic architecture knowledge in modeling, while the practical implementation of a design depends on its complexity and the architect's familiarity with design patterns and techniques. The structure of the 4x6 Method forces the architect to systematically think about the design intent and concerns, fostering gradual design implementation. To quickly introduce the tool and 4x6 concepts to the architects' community, coaching is done on a train-the-trainer manner, with mentoring and support by the central CA architecture team.

Starting in March 2011, a large number of architecture teams have applied the 4x6 Method and associated CAM tool. During the period of March 1, 2011 to October 21, 2011, a total of 53 distributed enterprise-grade products were involved in architecture reviews. Of them, 20 projects were *Versions* (implying a potentially more significant architectural effort) and the other 33 were *Releases* (large development efforts extending a current architecture).

Of the 53 projects aforementioned, 27 adopted the 4x6 Method. Of these, seven were Versions and the remaining 20 were Releases. From the start of a project to its review checkpoint took an average of 5.2 effort-months for a Version (11 month maximum – 2 month minimum), and a slightly higher average of 6.0 effort-months for a Release (12 month maximum – 2 month minimum).

24 projects kept the old style of architecture documentation. Of these, 12 were Versions and 12 were Releases. The corresponding efforts to reach the review checkpoints for these projects were in average 6.5 (13 Month Maximum – 1 month minimum, for an exceptionally small effort) and 7.7 effort-months for a Version and Release, respectively. The remaining two projects produced each a unique document. Both of these were Releases and took 8 and 9 months.

Of the 27 projects using the 4x6 Method as the foundation for their architecture, 12 were told no formal review was required because the reviewers understood the approach as documented. 11 were Releases and 1 was a small Version. The 15 projects formally reviewed had an average of three significant comments as a result of the review. Among those projects using the old style for documenting their architecture, 12 projects formally reviewed in a meeting had an average of 7 significant comments.

Reviewers reported less time was needed to prepare for the review session when dealing with architecture documents based on the 4x6 Method, and a better understanding of the designs was possible, when compared with the old approach to architecture modeling and documentation. The reviewers also remarked on the consistency of the models across projects, and how this aided comprehension and identified areas that might have not been discussed otherwise.

Project review meetings, on average, were 90 minutes long for a 4x6 Method based project, compared to 120 minutes (or more, in a few cases) for the old method.

5 Conclusion

In this paper we presented the 4x6 Tiered Architecture Method, the core of the CA Four Architecture views model-driven architecture methodology. Its use enables the capture of business, functional, logical and deployment views to maintain control over architecture evolution. In particular, design intent and architecture directives are captured within the 4x6 logical view. The resulting view can be used to state design goals and process, evolution steps, design rationale, as well as recommend best practices for structural composition of an enterprise product.

The ability to superimpose or integrate architecture elements requires them to be structured using identical format, templates and tools that foster collaboration and content reuse. In our case, the six logical layers and the use of design patterns are the methods, the CAM modeling tool is the means, and the component catalog serves to maintain and share information.

Both the 4x6 Method and CAM tool were tested and verified by practitioners on the design of real products, passing the corporate architecture review board with flying colors. By employing the systematic thinking and formal modeling of the architecture concerns, as highlighted by the supporting methodology and tools, the 4x6 approach forced architects to consider the use of design patterns, change the structural layout, and consider component implementation by third-party technology.

References

1. Kruchten, P.: The rational unified process: An introduction, 3rd edn. Addison-Wesley Professional, Boston (2003)
2. Rising, L.: The benefit of patterns. IEEE Software 27(5), 15–17 (2010)
3. Maiden, N.: Service design: It's all in the brand. IEEE Software 27(5), 18–19 (2010)
4. Hadar, E., Silberman, G.M.: Agile architecture methodology: Long term strategy interleaved with short term tactics. In: Companion to the 23rd ACM SIGPLAN Conference on Object-Oriented Programming Systems Languages and Applications, pp. 641–652. ACM, Nashville (2008)
5. Silverstein, D., Samuel, P., DeCarlo, N.: The innovator's toolkit: 50+ techniques for predictable and sustainable organic growth. Wiley, Hoboken (2008)
6. Buschmann, F., Henney, K.: Five considerations for software architecture, Part 1. IEEE Software 27(3), 63–65 (2010a)
7. Buschmann, F., Henney, K.: Five considerations for software architecture, Part 2. IEEE Software 27(4), 12–14 (2010b)
8. Burbeck, S.: Application programming in Smalltalk-80: How to use model-view-controller (MVC) (1992), `http://st-www.cs.illinois.edu/users/smarch/st-docs/mvc.html` (retrieved)
9. Bosch, J.: Toward compositional software product lines. IEEE Software 27(3), 29–34 (2010)
10. Ferguson, D.F., Hadar, E.: Constructing and evaluating supply-chain systems in cloud-connected enterprise. In: Moinhos Cordeiro, J., Virvou, M., Shishkov, B. (eds.) Proceedings of the International Conference on Software and Data Technologies, ICSOFT 2010, pp. 69–76. SciTePress, Athens (2010)
11. Sherman, S., Hadar, I., Hadar, E., Harrison, J.: Architecture documentation for agile development. Paper Presented at SATURN 2011, San Francisco, California (May 2011)
12. Booch, G.: An architectural oxymoron. IEEE Software 27(5), 96 (2010)
13. Buschmann, F.: On architecture styles and paradigms. IEEE Software 27(5), 92–94 (2010)
14. Hadar, E., Perreira, M.: Web services variation façade – Domain specific reference architecture for increasing integration usability. In: IEEE International Conference on Web Services, pp. 1207–1211. IEEE Computer Society, Salt Lake City (2007)

A Layered Architecture for Enterprise Data Warehouse Systems

Thorsten Winsemann[1,2], Veit Köppen[2], and Gunter Saake[2]

[1] SAP Deutschland AG & Co. KG, Großer Grasbrook 17, 22047 Hamburg, Germany
Thorsten.Winsemann@t-online.de
[2] Otto-von-Guericke Universität, Universitätsplatz 2, 39106 Magdeburg, Germany
{Veit.Koeppen,Gunter.Saake}@ovgu.de

Abstract. The architecture of Data Warehouse systems is described on basis of so-called reference architectures. Today's requirements to Enterprise Data Warehouses are often too complex to be satisfactorily achieved by the rather rough descriptions of this reference architecture. We describe an architecture of dedicated layers to face those complex requirements, and point out additional expenses and resulting advantages of our approach compared to the traditional one.

1 Characteristics of an Enterprise Data Warehouse

A *Business Data Warehouse* (BDW, [1]) is a Data Warehouse (DW) to support decisions concerning the business on all organizational levels. It covers all business areas, such as logistics, finance, and controlling. We define Enterprise Data Warehouses (EDW) as subspecies of BDW systems, which not only cover all activities of an enterprise, but are an important basis for applications, such as Business Intelligence, planning, and Customer Relationship Management. An EDW collects and distributes huge amounts of data from a multitude of heterogeneous source systems. It has to provide a single version of truth for all data of the company. That means that there is one common view on centralized, accurate, harmonized, consistent, and integrated data at a given point of time. The range of use is often world-wide, so that data from different time-zones have to be integrated. Frequently, 24x7-hours data availability has to be guaranteed, facing the problem of loading and querying at the same time. In addition, there are high requirements to the data: ad-hoc access, near real-time availability, high data quality, and the need for very detailed and granular data with a long time horizon. Moreover, new or changing requirements for information have to be flexibly and promptly satisfied, and, last but not least, the data access has to be secured by a powerful authorization concept.

These significant requirements for an EDW also necessitate enhancements and refinements to the DW architecture, compared to the traditional one, which is outlined in this paper. Sections 2 and 3 describe the traditional and the layered DW architecture and show a comparative example of both approaches. In Section 4, we discuss additional expenses and resulting advantages of the layered architecture approach in detail. Section 5 concludes our paper and gives an outlook on future work.

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 192–199, 2012.

2 Traditional Data Warehouse Reference Architectures

In literature, several DW architectures are described [e.g., 2, 3, 4], and DW systems are mostly modeled with three to five areas as *reference architectures* [cf. 5, 6, 7, 8]; see Figure 1 for a simplified model. Data are extracted from data sources into the staging area, where they are transformed to the common DW schema. Afterwards, data are loaded into the basis database of the DW, where they are stored at the required level of granularity. Based on this, data marts are built; that means redundant copies of data that are defined according to the users' requirements, for instance for analysis. The operational data store is used for special business cases, such as near real-time reporting. In other words, one can define three aspects of data handling: data acquisition, data processing, and data provision.

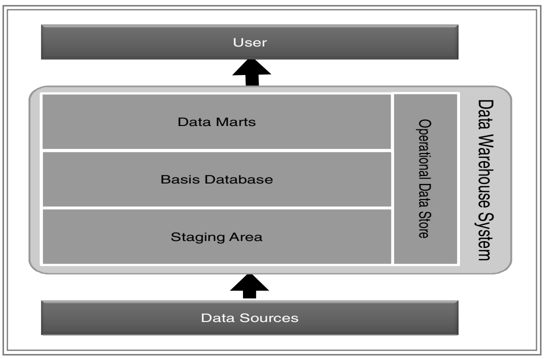


Fig. 1. DW Reference Architecture

In Figure 2, we illustrate this theoretical approach with an example from the business area of sales and distribution in general, and the sales documents of orders and invoices in particular. In an Enterprise Resource and Planning system (ERP), data are kept in four different tables: sales order and sale invoice, header and item data. Data are selected and joined in ERP, extracted to the DW, transformed (i.e., cleansed, harmonized, etc.) in the staging area and loaded into the basis database. Then, data marts are filled that are designed for fast analysis' access.

However, practical experiences show that this rough architectural model is insufficient in several aspects. Data access for analyses does not only take place at the data mart level, but is also applied to the basis database – the boundaries are fluid [9]. ETL (Extraction, Transformation, and Loading) processes in this model consist of different steps of data processing and reach from data source via staging area into the basis database [e.g., 8, 10]. In practice, data marts are often used as sources of the basis database, and data are not only transformed (e.g., enriched with attributes) in the staging area, but also in the basis database and data mart area. When accessing data for analyses, they are usually also transformed (e.g., aggregated or averaged). Therefore, transformation cannot be considered as a process which ends when data are stored in the basis database. It applies over several layers and in all areas of a DW system: data acquisition, data processing, and data provision. Moreover, the complexity of data processing is not expressed adequately – although it is one of the biggest problems when building and operating a DW [6]. Data transformation processes with more than five levels and execution times of several hours are not rare within a productive EDW environment.

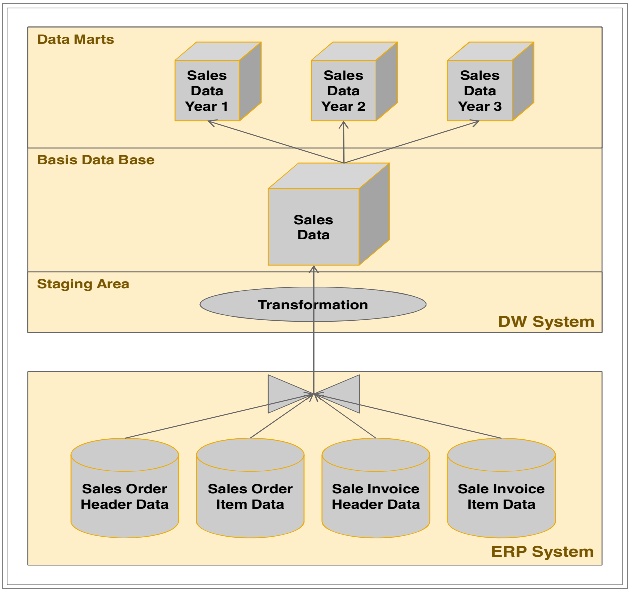


Fig. 2. Data-Flow-Example in a Reference Architecture

3 The Layered Architecture Approach

A layered architecture improves the reference architecture approach with respect to a comprehensive and ordering collection and classification of data transformation. It describes levels of data alteration within a DW from data acquisition to data processing and data provision. Layers become more detailed, dedicated, and purposeful herein.

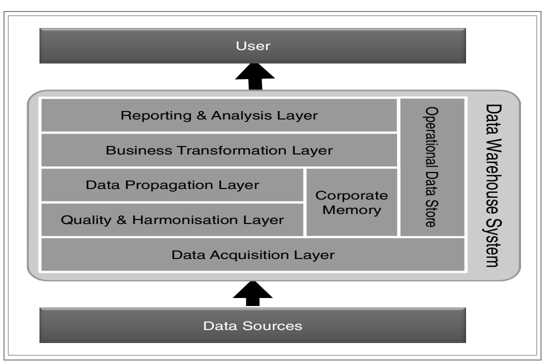


Fig. 3. Layered, Scalable Architecture for EDW [11]

A good example for such architecture is SAP's "Layered, Scalable Architecture (LSA)" for Enterprise Data Warehousing, shown in Figure 3. It is defined as a reference for designing architectures - not limited to SAP-based systems - according to individual and actual requirements.

Extracted data are stored immediately in tables of the *Data Acquisition Layer*, without any transformations. Thereby, the source system is decoupled immediately and no longer strained after successful data transfer.

Data in the acquisition layer are deleted after a certain time when they are successfully loaded to all dedicated data targets. Yet, if it is stored in the long term, it can be defined as *Corporate Memory* and enables access to all loaded data. For easier re-use, some administrative information can be added (e.g., data origin and timestamp). Hence, data are available in the DW, even when they are already deleted or changed in the source system; usually, data cannot be recovered and are lost. Furthermore, re-loading data causes organizational problems, such as necessary down-times. It is advisable to extract not only actual necessary data from one source, but all possibly useful data. Thus, the data base represents a reservoir for unpredictable requirements – without re-modeling data flows from source systems and extracting data that are missing in the DW. By definition, the use is rare, so that storage mediums can be slower and cheaper ones.

In the *Quality & Harmonisation Layer*, necessary integration of data is done; this includes syntactical and semantical integration, schema mapping, consolidation, cleansing, quality-checks, data validation, and de-duplication [cf. 10, 12]. The result is harmonized and integrated data, stored in the overlying layer.

The *Data Propagation Layer* contains harmonized, integrated data. It represents the data's single point of truth as the basis for any further usage. Therefore, it has to be the layer, where a clear, common, and company-wide understanding of the meaning of the data is defined (e.g., "what really does working day mean?", "how is revenue defined?"). Data are kept as granular as possible with an adequate time horizon and free of any business logic. Data are loaded only once into the DW, and are deployed several times, so that further system load, due to redundant extraction, staging, and storage of data, is avoided.

The absence of business logic in the propagation layer offers utmost flexibility for the use of data in the upper layers. Data are transformed regarding the business needs in the *Business Transformation Layer*. Such business needs can be currency conversions with month's end exchange rate, computation of key indicators, or business-defined combination of data from different areas. Our example shows a simple transformation of merging document header and item data. However, one can imagine complex combinations of data from different business areas, such as purchasing and inventory management, or logistics and finance.

The main task of the *Reporting & Analysis Layer* is to offer data as "ready-to-use". Due to a better query performance or complex transformation of data, a further materialization can be necessary (as illustrated for the sales data years 1 – 3 in Figure 4). Yet, the data can also be read from layers below and are presented as a virtual cube for reporting (as shown for "Special Sales Data").

The *Operational Data Store* – which is not implemented in our example – is mainly dedicated for special needs of application-specific, granular data with near real-time availability.

A rough mapping of these layers to the three DW's aspects of data handling can be done: data acquisition is covered within the Data Acquisition and the Quality & Harmonisation Layer, data processing in the Data Propagation and Business Transformation Layer, and data provision in the Reporting & Analysis Layer.

This architecture is a conceptual matter. Each layer represents a designated area, in which data are changed with respect to their actual format and their dedicated usage. The integration of each layer into the individual DW architecture mainly depends on the need for such changes. For instance, a data set does not have to be lifted to the data mart layer if it is already usable on a lower one. Further aspects result from design decisions or DW architects' preferences; for example whether to build a dedicated corporate memory or to combine it with the acquisition layer. The assignment of transformations and layers illustrates the increase of the *data's excellence* – the added value of the data that is reached on each layer.

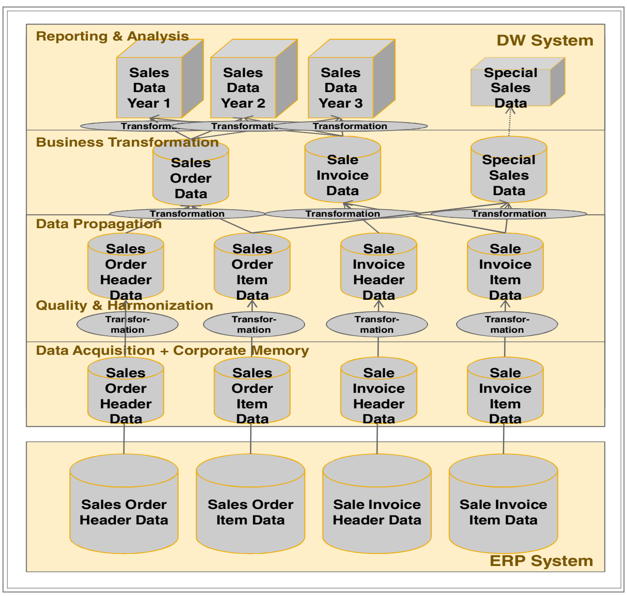


Fig. 4. Data-Flow-Example in a Layered Architecture

We make the difference to the traditional architecture clearer by adapting the data flow shown in Figure 2 to the layered architecture in Figure 4. Data are selected in ERP and extracted to the DW system, usually one data flow per source. In the DW, data are stored unchanged in the acquisition layer, which is also defined as corporate memory. Subsequently, the data are cleaned, quality checked, and harmonized. Afterwards, they are stored into tables in the propagation layer, free of business-related changes. Such changes are made in the business transformation layer, where the two data sets ("Sales Order" and "Sale Invoice") are merged and a "Special Sales" set is created, in which order and invoice items' data are combined. On top, several data marts (one for each year) are defined for better performance. The "Special Sales Data" mart represents a virtual access point without data and clarifies that reporting is possible on data of each layer.

4 Architectural Differences in Detail

Comparing Figures 2 and 4, the higher complexity of the layered architecture is obvious. It initially covers complete business areas and combines data according to actual requirements in a subsequent step. This leads to a higher volume of data to be extracted to the DW. Besides, more conceptual work is necessary than in the traditional approach of defining the DW on base of the actual users' requirements. Hereby, duration and costs of the implementation project are affected. However, there are several advantages which justify this kind of architecture. We illustrate these advantages by exemplarily present some common activities in an EDW, such as change of transformation rules, change of data, and need for new data.

The transformation of data loaded into the DW can be erroneous: incorrect computation of key indicators, currency conversion based on over aged rates, or assignment of countries to regions has changed. In the traditional architecture, data are usually not available in a re-usable format and must be re-loaded from source systems. In contrast, the propagation layer holds data in a format that enables a smooth re-building into the overlying levels. Occasionally, data must be combined differently due to new business requirements; for instance, sales regions are restructured or new key indicators must be computed. Again, data re-loading is usually done in traditional architecture, whereas computation can be started from the propagation layer in the layered architecture. Often, business requires data that are not included in the initial concept of the DW's design. Even in case these new data are part of already connected sources, they are not included in the extraction. Hence, the only way in the traditional architecture is an enhancement of the data flow from the source to the DW, followed by a complete new loading. As pronounced above, the approach of the layered architecture is to get all possibly relevant pieces of information into the DW when a new source is extracted. Therefore, the probability is high, that such data are already kept in the corporate memory or in the propagation layer, including previous data.

The above mentioned examples clarify that the layered architecture offers fast ways to satisfy new or changing needs for information, which are frequent use cases. Furthermore, defining data in the propagation layer as single point of truth supports a common, company-wide view on integrated and trustable data with respect to data governance. Based on this, even local or departmental data marts, or such created for special purposes, contain reliable information. Moreover, the data stream of loading and using for reporting is decoupled. By this means, it compensates extraction of data from different time zones, which are released into the DW's basis database at one defined point in time. Continuing, the data availability with detailed previous data is a highlight, too. Moreover, the layered architecture offers easy possibilities of scalability, especially when a proper naming convention has been established. Due to openness and flexibility of the layer concept, the system can easily be enhanced by integrating data streams or copying of applications. Here, the initial additional expenses pay off. This becomes necessary, whenever data from another region, country, business area or company have to be absorbed into the EDW. For example, a new business area is defined in ERP. As data extraction is not limited to existing ones, such data are transferred to the DW instantly. As there is no business-related data modification up to the propagation layer, the DW is not affected by this enhancement until modeling is necessary in the business transformation and reporting layer.

[2] points out several "metrics for assessing architecture success", such as *information quality* (i.e., accuracy, completeness, and consistency of data) and s*ystem quality* (i.e., flexibility, scalability, and integration of the system). In view of EDW's complexity, a layered architecture offers good means to cope with it.

5 Conclusion and Outlook

We introduce the main characteristics of an EDW and explain the traditional reference architecture for DW systems. We illustrate the layered architecture on base of SAP's "Layered, Scalable Architecture", which represents a common and field-tested example of such an approach. It defines layers as designated areas, in which data are changed with regard to their actual format and their dedicated usage. Compared to the traditional approach, building a DW with a layered architecture needs more conceptual work and additional expenses. Certainly, it enables a much clearer, dedicated assignment of data transformation to each layer. The advantages of the layered architecture outbalance the traditional one, regarding consistency of data, flexibility, reusability, and scalability – especially with respect to EDW's complexity. This shall not only be in mind when building a new DW, but even lead to think about re-designing existing ones.

Future work will include detailed, exemplary illustrations as well as validations of the described layered approach by comparing it to the traditional reference architecture on base of SAP and non-SAP DW platforms.

References

1. Devlin, B.A., Murphy, P.T.: An architecture for a business and information system. IBM Systems Journal 27(1), 60–80 (1988)
2. Watson, H.J., Ariyachandra, T.: Data Warehouse Architectures: Factors in the Selection and the Success of the Architectures (2005), `http://www.terry.uga.edu/~hwatson/DW_Architecture_Report.pdf` (call April 01, 2012)
3. Hajmoosaei, A., Kashfi, M., Kailasam, P.: Comparison plan for data warehouse system architectures. In: 3rd ICMiA Proceedings, pp. 290–293 (2011)
4. Golfarelli, M., Rizzi, S.: Data Warehouse Design: Modern Principles and Methodologies. McGraw-Hill (2009)
5. Poe, V.: Building a Data Warehouse for decision support. Prentice Hall (1996)

6. Chamoni, P., Gluchowski, P. (eds.): Analytische Informationssysteme. Springer (2006)
7. Muksch, H., Behme, W. (eds.): Das Data-Warehouse-Konzept. Gabler (2000)
8. Zeh, T.: Referenzmodell für die Architektur von Data-Warehouse-Systemen (Referenzarchitektur) (2008), `http://www.tzeh.de/doc/gse-ra.ppt` (call April 01, 2012)
9. Lehmann, P.: Meta-Datenmanagement in Data-Warehouse-Systemen. PhD Thesis, University of Magdeburg (2001)
10. Leser, U., Naumann, F.: Informationsintegration. dpunkt (2007)
11. SAP AG: Layered, Scalable Architecture (LSA) for BW, Training Material (2009)
12. Lehner, W.: Datenbanktechnologie für Data-Warehouse-Systeme. dpunkt (2003)

Graph-Based Pattern Identification from Architecture Change Logs

Aakash Ahmad, Pooyan Jamshidi, and Claus Pahl

Lero - The Irish Software Engineering Research Center
School of Computing, Dublin City University, Ireland
{ahmad.aakash,pooyan.jamshidi,claus.pahl}@computing.dcu.ie

Abstract. Service-based architectures have become commonplace, creating the need to address their systematic maintenance and evolution. We investigate architecture change representation, primarily focusing on the identification of change patterns that support the potential reuse of common changes in architecture-centric evolution for service software. We propose to exploit architecture change logs - capturing traces of sequential changes - to identify patterns of change that occur over time. The changes in the log are formalised as a typed attributed graph that allows us to apply frequent sub-graph mining approaches to identify potentially reusable, usage-determined change patterns. We propose to foster the reuse of routine evolution tasks to allow an architect to follow a systematic, reuse-centered approach to architectural change execution.

Keywords: Service-driven Architecture, Change Patterns, Evolution.

1 Introduction

Software architecture represents the global system structure for designing, evolving and reasoning about the configurations of computational components and their interconnections at higher abstraction levels. Service-Oriented Architecture (SOA) is an architectural approach that models business processes as technical software services. Once deployed, continuous change in business and technical requirements leads towards frequent maintenance and evolution in service systems [12]. In order to accommodate recurring changes in the SOA lifecycle, the solution lies in developing processes, frameworks and patterns that enable change reuse for architectural evolution of service software [12].

We have been working on the 'Pat-Evol' project [2, 3] that aims at supporting pattern-driven reuse in architecture-centric evolution for service-driven software. Based on the taxonomy of software change [1], we believe that a systematic investigation of architecture change history could help us to discover sequences of recurring change that occur over time. Recurring changes can be exploited to identify change patterns that support a generic, potentially reusable solution to recurring architecture evolution problems. Therefore, we focus on change representation and its operationalisation by maintaining an architecture change log -

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 200–213, 2012.

tracing each individual change - for our case studies. The change log keeps a sequential history (as the 'post-mortem' data) of architectural changes, providing us with an empirical foundation to identify patterns of change.

Although a recent emergence of evolution styles [5, 13, 6, 7] promotes the 'build-once use-often' philosophy in architecture and process evolution, it falls short of addressing frequent demand-driven process-centric changes [15, 16] that are central to maintenance and evolution of SOAs. This motivates the needs to systematically investigate architecture change representation that goes beyond frequent addition or removal of individual components and connectors to operationalise recurrent process-based architectural changes.

The proposed solution is based on formalising architectural changes from the logs as a typed attributed graph [8] that provides formal semantics with its node and edge attribution to operationalise architectural changes [9]. We utilise frequent sub-graph mining [4] techniques to not only identify the exact instances, but also inexact matches where only central pattern features suffice for identification. The scalability of solution beyond manual analysis is supported with a prototype 'G-Pride' (Graph-based Pattern Identification) that facilitates automation and parameterised user intervention for pattern identification process. We believe, a continuous experimental identification of patterns is the first step towards facilitating the architect(s) to capitalise on a reuse-centered approach to systematically accommodate recurring changes in existing software.

This paper is organised as follows. A formal specification for the change pattern(s) is presented in Section 2, followed by an overview of the proposed solution in Section 3. We elaborate on graph-based pattern identification in Section 4 and its evaluation in Section 5. In order to justify the overall contribution, related work is presented in Section 6 that is followed by conclusions.

2 Change Pattern

In change logs, we observed that the operationalisation of individual changes represent a parameterised procedural abstraction. This helps us to define change pattern as a *generic, first class abstraction (that can be operationalised and parameterised) to support potentially reusable architectural change execution.* We present a formal description of change pattern in terms of a meta-model of its constituent elements in Figure 1a along with its properties in Figure 1b.

2.1 Pattern-Based Architecture Evolution

We model pattern-based evolution $PatEvol = < SArch, OPR, CNS, PAT >$ as 4-tuple with element inter-relationships in Figure 1a as explained below.

1. **Service Architecture (SArch)** refers to the architecture elements to which a pattern can be applied during change execution. We utilise attributed typed graphs [8] that provide formal syntax and semantics with its node and edge attribution to model typed instances of architectural elements. We use

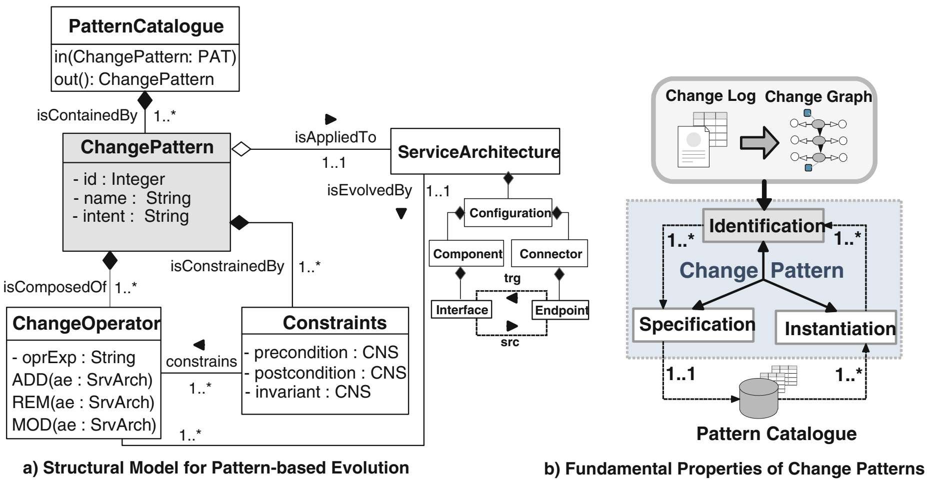


Fig. 1. Model Representation for Reusable Architecture Evolution

the Graph Modeling Language (.GML) for an XML-based representation of architectural instances. The architectural model is consistent with the Service Component Architecture specifications that include configurations (CFG) among a set service components (CMP) as the computational entities that are linked through connectors (CON), in Figure 1a. The modeling is restricted to service-based architectures that only support composition or association type dependencies among service composites. Thus, the structural integrity of architecture elements and consistency of pattern-based change beyond this architecture definition is undefined.

2. **Change Operator (OPR)** represents operationalisation of changes that is fundamental to architectural evolution. Our analysis of the log goes beyond basic change types [1] to define a set of atomic and composite operations enabling structural evolution by adding (ADD), removing (REM) and modifying (MOD) the architecture elements (AE). An inherent benefit of graph-based modeling is the support for architectural evolution by means of graph transformations. More specifically, during change execution the operations could be abstracted as graph transformation rules (in our case supported by XML transformations using XSLT). This enables a fine-granular operationalisation $OPR(ae \in AE)$ to preserve the compositional hierarchy of architecture elements during change execution with:
- *Atomic Change Operations:* enable fundamental architectural changes in terms of adding, removing or modifying the service operation (OPT), service interface (INF), connector binding (BIN), connector endpoint (EPT) and configuration interface ($cfgINF$).
- *Composite Change Operations:* are a set of atomic change operations, combined to enable composite architectural changes. These enable adding, re-

moving or modifying the components (CMP), connectors (CON) and configurations (CFG) with a sequential composition of architectural changes.
3. **Constraints (CNS)** refer to a set of pattern-specific constraints in terms of pre-conditions (PRE) and post-conditions ($POST$) to ensure consistency of pattern-based changes. In addition, the invariants (INV) ensure structural integrity of individual architecture elements during change execution.
4. **Change Patterns (PAT)** represents a recurring, constrained composition of change operationalisation on architecture elements that is specified as: $PAT_{<id,\ name>} : PRE(ae_m \in AE) \xrightarrow{INV(OPR_n(ae_m \in AE))} POST(ae'_m \in AE)$. Constraints enforcement on operational composition ensures structural integrity of architecture elements during pattern-based change execution.

A **pattern catalogue (CAT)** refers to a template-based repository infrastructure to facilitate automated storage (in: once-off specification) and retrieval (out: multiple instantiation) of change patterns during evolution.

2.2 Fundamental Properties of Change Pattern

In addition to the meta-model, the fundamental properties of change pattern are presented in Figure 1b. In order to capitalise on pattern-driven reuse, these properties support our argument about change pattern as a generic solution that can be i) identified as recurrent, ii) specified once and iii) instantiated multiple times to support potentially reusable architectural change execution.

- *Identification:* aims at an empirical investigation about the history of architectural changes to identify recurring sequences of change that occur over time. The motivation for architectural change investigation is to discover and analyse real changes (i.e., not any assumed data sets) by extracting the implicit evolution-centric knowledge from change logs, our focus in this paper.
- *Specification:* after identification, it is vital to provide a consistent (once-off) specification for the collection of identified change patterns as pattern catalogue. A template-based specification facilitates flexible querying and retrieval whenever a need for pattern usage arises.
- *Instantiation:* in order to realise the concept of pattern-driven change execution, it allows instantiation of appropriate pattern(s) from its abstract specification to promote the concept of 'specify-once, instantiate-often' approach during architecture evolution.

3 Automating Change Pattern Identification

We investigated architectural changes empirically - analysing change representation - to discover recurrent sequences in a change log. Therefore, we have based the identification of patterns on the analysis of changes for two service-based system evolutions we recorded in the log. These include an Electronic Billing Presentment and Payment (EBPP) system whose specifications are published by NACHA and an on-line Tour Reservation System (TRS). We propose a three-phase approach to identify architecture change patterns, in Figure 2.

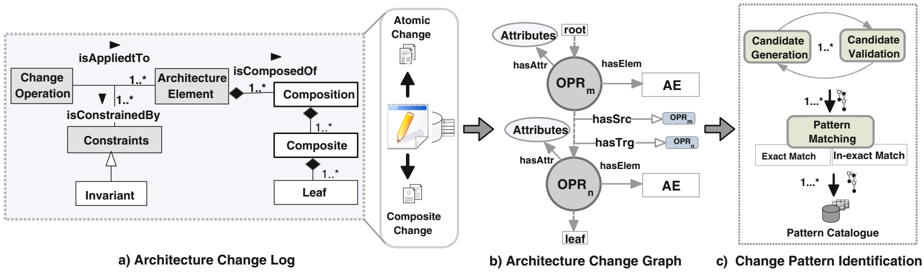


Fig. 2. An Overview of the Proposed Pattern Identification Process

3.1 Maintaining the Architecture Change Log

As the initial step, we follow [1] to record the architecture 'change history'. We use a centrally manageable repository to record sequential architectural changes that are constrained by a set of invariants. We expand on the idea of process change logs from [16] and tailor it to record each individual architectural change as the log tuple. The structural view for the change log is presented in Figure 2a that acts as the foundation to identify change patterns with specific frequency thresholds. While analysing the change operationalisation, each individual change from the case studies is manually recorded in the log that currently comprises of more than a couple of thousands of changes. (i.e., $OPR > 2000$). The primary benefits of this approach included:

- Maintaining the traces of evolution in an updated central repository.
- Analytical support with searching and querying concrete instances of change.
- Experimental analysis of change representation, patterns identification etc.

In Figure 2, the structure of change log maintains the compositional hierarchy of elements. For example, every service component (Composition) must contain at least one or more interfaces (Composite) containing one or more operations (Leaf), while connectors must have binding that contain sn endpoint. We are specifically interested in analysing recurrent sequences that exhibit process-centric changes (e.g. integration, replacement, decomposition etc.) that are central to SOA evolution, as composite changes based on addition or removal of individual components and connectors.

3.2 Graph-Based Formalisation of Architectural Changes

We formalise the architectural changes in the log as an attributed graph (AG) with its nodes and edges typed over an attributed typed graph (ATG) [8] using an attributed graph morphism $t : AG \rightarrow ATG$ as indicated in Figure 3. The ATG provides formal semantics to represent atomic and composite changes with visualisation, efficient searching and analysis of sequential changes in the log. However, we are specifically interested in exploiting frequent sub-graph mining to identify recurring sequences as potential change patterns.

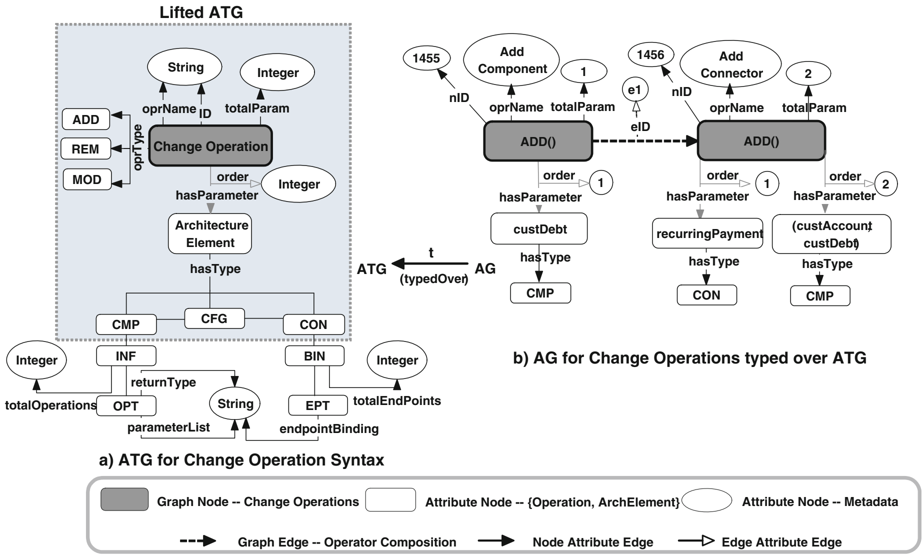


Fig. 3. Attributed Graph to Formalise Architecture Change Operationalisation

Lifting the Change Graph - Sequential Composition: In the change log, analysing an individual change lacks the required abstraction to exploit the recurrent process-centric changes. Furthermore, taking into consideration the granularity of architectural changes (OPR in Section 2) there does not exist a unified representation for architectural evolution that satisfies the needs for all stake-holders' view. For example, a software developer might be more interested in analysing the modification of a specific operation's signature and their semantics, while the architect would be exclusively concerned about the affected component-level interconnections. The possible views could be virtually unlimited depending on any specific evolutionary perspectives. However, in this paper, instead of focusing on atomic changes we focus on sequential composition identification that exhibits process-centric aspects of change in terms of integration, replacement, decomposition of elements. Therefore, we apply graph lifting [11] to enable projection onto higher-level architectural composites that include configurations, components and connectors, hiding low-level atomic changes.

Creating the Change Session Graph: Once the graph is lifted, we provide a utility method as sessionGraph(uID, strTime, endTime) to automatically create the change graph based on a particular change session in the log. The change session is determined by the identification of the user (uID) who applied the change(s) in a specific time interval ($endTime$ - $strTime$). For experimental purposes, we consider all the changes in the log as a single session to extract the attributes of change instances that, we generate the lifted change graph (Figure 3a - dotted blue square) with a concrete represention using the Graph Modeling Language (.GML) format. The result is a directed graph representing sequential composition of change operationalisation, illustrated in Fig. 3b.

For clarity of presentation, some additional attributes (like date, time, committer of change etc.) from the actual graph are hidden to focus on the sequencing of operations on architecture elements. The attributed graph morphism $t : AG \rightarrow ATG$ is defined over graph nodes with $t(MetaData) = \mathsf{ChangeData}$ that results in $t(ChangeOperation) = \mathsf{ADD()}$, $t(ArchitectureElement) = \mathsf{custDebt}$, $\mathsf{recurringPayment}$, $\mathsf{custAccount}$ and $t(hasType) = \mathsf{CMP}$, CON in Figure 3. The change operationalisation as a typed attributed graph is expressed as 5-tuple: $G_C = \; < N_G, N_A, E_G, E_{NA}, E_{EA} >$, with:

1. **Graph Nodes** $N_G = \{n(g_i)|i = 1, ..., k\}$ is the set of graph nodes. Each node represents a single change operation (i.e., add a component, remove a connector etc.), where t(N_G) = ADD(), REM(), MOD().
2. **Attributed Nodes** $N_A = \{n(a_i)|i = 1, ..., l\}$ is the set of attribute nodes. Attribute nodes are of two types: i) attribute nodes with change metadata, e.g. change operation, name, number of parameters and ii) attribute nodes representing architecture elements (and their compositions) e.g. configuration, component, connector etc, where t(N_A) = (AE : hasType).
3. **Graph Edges** $E_G = \{e(g_i)|i = 1, ..., k - 1\}$ is the set of graph edges which connect source $n(g)_{src}$ and target $n(g)_{trg}$ nodes. It represents the sequencing of change operations in the graph, where t(E_G) = eID($N_{Gi_{src}}$, $N_{Gi_{trg}}$).
4. **Node Attributed Edges** $E_{NA} = \{e(na_i)|i = 1, ..., m\}$ is the set of node attribute edges which join a graph node n(g) to an attribute node n(a), where t(E_{NA}) = nodeAttr(String), e.g. *nID, oprName, totalParam.*
5. **Edge Attributed Edges** $E_{EA} = \{e(ea_i)|i = 1, ..., n\}$ is the set of edge attribute edges which join a node attribute edge e(na) to an attribute node n(a), where t(E_{EA}) = edgeAttr(String), e.g. *eID, eName.*

For example in Fig. 3b, the attributed graph represents two change operations extracted from EBPP architectural changes. It illustrates the addition of a new service component (custDebt hasType CMP) that is connected to an existing component (custAccount hasType CMP) with a connector (recurringPayment hasType CON). The graph nodes are linked to each other using graph edges e(g) having edge id (*e1*) along with the ids of its source and target nodes (*1455, 1456*) representing the applied sequence of change operations.

4 Graph-Based Identification of Change Patterns

Once architectural changes in the log are formalised as an architecture change graph G_C, the final step involves graph-based identification of change patterns. We exploit one of the classical approaches for pattern mining with sub-graph isomorphism [4] from recurring sub-graphs G_P to G_C, where $G_P \subseteq G_C$.

4.1 Properties and Types of Change Sequences

Operationalising the change representation is particularly beneficial to define sequential composition of change operations on architecture elements. This allows

Table 1. Change Sequences (S_x and S_y) as Extracted from the Change Log

	Sequence 1 ($\mathbf{S}_x$)		**Sequence 2 ($\mathbf{S}_y$)**		
cID	**OPR**	**Architecture Elements**	**cID**	**OPR**	**Architecture Elements**
77	REM()	getInvoice ∈ CMP	312	ADD()	paymentType ∈ CMP
78	ADD()	custBill ∈ CMP	313	REM()	custPayment ∈ CMP
79	REM()	payInvoice ∈ CON	314	REM()	getBillData ∈ CON
80	ADD()	payBill ∈ CON	-	-	-

us to abstract the individual changes into a sequence of recurring change operations representing potential patterns determined by the following properties.

In order to exemplify the properties, Table 1 represents two change sequences (S_x and S_y) extracted from the change log. The sequences contain change id (*cID*), change operation (*OPR*) and the affected architecture element (*AE*). Note, that for space reasons we do not explicitly represent the parameters for connectors as they are insignificant during sequence matching. Sequence 1 (S_x) represents the replacement of the existing component getInvoice with custBill and the corresponding connectors payInvoice, payBill. Sequence 2 (S_y) represents the addition of a new component paymentType that is followed by removal of an existing component custPayment and a connector getBillData.

Type Equivalence (TypeEqu) refers to the equivalence of two change operations given by the utility function $TypeEqu(OPR_1(ae_i : AE), OPR_2(ae_j : AE)) : returns < boolean >$. It depends on the type of change operation and the architecture element for a change operation to categorised as type equivalent (return true) or type distinct (returns false). For example, the change operation $REM(getInvoice \in CMP)$ is only equivalent to $REM(custPayment \in CMP)$ and TypeEqu(77, 313) returns *true*, as in Table 1.

Length Equivalence (LenEqu) refers to the equivalence of length of two change sequences where length of a change sequence is defined by the number of change operation contained in it. It is given by the function $LenEqu(S_x, S_y) : returns < integer >$. Therefore, the length equivalence of two change sequences S_x and Sy is determined by the numerical value (0 imples $S_x == S_y$, -n implies $S_x < S_y$ by n nodes and +n implies $S_x > S_y$ by n nodes). For example, in Table 1 the length of $S_x > S_y$ by one operation so TypeEqu(S_x, S_y) returns 1.

Order Equivalance (OrdEqu) refers to the equivalence in the order of change operations of two sequences. Analysing the change log based on a given change session, we observed that it is normal for same user to perform similar changes using different sequencing of change operations. The semantics and impact of change operation remains the same even if sequencing of change operations is varied. It is given by the function $OrdEqu(S_x, S_y) : returns < boolean >$. We distinguish different types of identified sequences, in Table 2.

- *Exact Sequence:* Two given sequences are exact subsequences if they match on operational types, length equivalence and the ordering of the change operations. In Table 1, S_x and S_y are not the exact sequences because in both the sequence length and the order of operation do not match.

Table 2. The Types of Identified Sequences in the Change Log

Sequence Type	TypeEqu	LenEqu	OrdEqu
Exact Sequence	true	0	true
Inexact Sequence	true	0	false
Partial Exact Sequence	true	$\pm$ n	true
Partial inexact Sequence	true	$\pm$ n	false

- *Inexact Sequence:* Two given sequences are inexact matching sequences if their operational types and lengths are equivalent, but order of change operation varies. In Table 1 S_x and S_y are not the inexact matching sequences as $S_x > S_y$.

- *Partial Exact Sequence:* Two given sequences S_x and S_y are partially exact such that (if $n > 0$ than $S_y \subset S_x$, or if $n < 0$ than $S_x \subset S_y$) - however, the types and ordering of the change operations in the matched sequences must be equivalent. In Table 1 S_x and S_y are not partial exact matching sequences as the order of operations in both the sequences do not match.

- *Partial Inexact Sequence:* Two given sequences S_x and S_y are partial and inexact if (if $n > 0$ than $S_y \subset S_x$, or if $n < 0$ than $S_x \subset S_y$); in addition, the operations within both sequences must be type equivalent, while the order of change operations in both sequences varies. In Table 1 S_x and S_y are partial inexact match. Although $S_x > S_y$, still $S_y \subset S_y$ as cID(77, 78, 79) matches cID(312, 313, 314) ($\mathsf{OrdEqu}(S_x, S_y)$ returns *true*).

4.2 Pattern Identification Process

The properties in Table 2 are vital to not only identify exact instances, but also inexact matches and possible variants of a change pattern where only some pattern features suffice for identification. We introduce the pattern identification problem[1] as a modular solution. This enables automation along with appropriate user intervention and customisation through parameterisation in Table 3 for pattern identification. We follow an apriori-based approach that proceeds in a generate-and-test manner using a Breadth First Search strategy during each iteration to i) generate and ii) validate pattern candidates from G_C and finally, (iii) determine the occurrence frequency of exact and inexact candidates in G_C.

Candidate Generation. The initial step of pattern identification generates a set of candidate sequences S_C from change graph G_C. A candidate consists of a number of nodes representing change operationalisation on architecture elements as a potential pattern depending on its occurrence frequency $Freq(S_C)$ in G_C. *Input(s)* is the change graph G_C and user specified minimum $minLen(S_C)$ and maximum $maxLen(S_C)$ candidate sequence lengths. *Output(s)* is a list of generated candidates $List(S_C)$ such that $minLen(S_C) \leq Len(S_{C_i}) \leq maxLen(S_C)$.

Candidate Validation. We observed that during candidate generation, there may exist some false positives in terms of candidates that violate the struc-

[1] The algorithms along with the developed prototype can be accessed at: `http://www.computing.dcu.ie/~aaakash/ChangePattern.html`

Table 3. List of User Specified Parameters for Pattern Identification

Parameter	**Description**
G_C	Change session graph created from change Log.
S_C	Candidate sequences generated from change graph: $S_C \subseteq G_C$.
G_P	Identified Pattern instance from change graph: $G_P \subseteq G_C$.
$Len(S_C)$	Candidate length representing number of change operations in S_C.
$minLen(S_C)$	Minimum candidate length that is specified by the user: $minLen(S_C) \leq Len(sc) : sc \in S_C$.
$maxLen(S_C)$	Maximum candidate length that is specified by the user: $maxLen(S_C) \geq Len(sc) : sc \in S_C$.
$Freq(S_C)$	Frequency threshold that is specified by the user for S_C to be identified as a pattern G_P.
$List(param : G_C)$	The list of candidates S_C or patterns G_P as $param \subseteq G_C$.

tural integrity of architecture elements when identified and applied as patterns. Therefore, it is vital to eliminate such candidates through validation for each generated candidate sequence *sc* against architectural invariants before pattern matching. *Input(s)* is a candidate $sc \in G_C$ (called from candidateGeneration(), each time a candidate is generated). *Output(s)* a boolean value indicating either valid (true) or invalid (false) candidate sequence *sc*.

Pattern Matching. The last step identifies exact and inexact instances of change patterns based on a user specified frequency threshold. This is achieved by structural matching using sub-graph morphism [4] among the nodes of $List(S_C)$ to corresponding nodes in G_C. *Input(s)* is a list of (validated) candidates $vList(S_C)$, specified frequency threshold $Freq(S_C)$ and G_C. *Output(s)* is a list of identified patterns consisting of pattern instance G_P and its frequency $Freq(G_P)$. A given candidate is an identified pattern (exact or inexact) if its frequency is greater or equal to the user specified threshold: $freq(G_P) \geq Freq(C_P)$.

5 Experimental Analysis and Illustration

The identified pattern types are generally classified as Inclusion, Exclusion and Replacement types depending on the impact of change as addition, removal or modification of elements from existing architecture.

5.1 Identified Pattern Instance - Component Integration

In Table 5, we present an identified instance of the co-related Inclusion pattern that is specified as Integrate (CNS, OPR, AE). Such a declarative specification facilitates the retrieval of appropriate patterns from a catalogue, consisting of the syntactical context that contains pattern pre- and post-conditions (CNS), the applied change operations (OPR) and the affected architecture elements (AE). The co-related Inclusion pattern aims at *integration of mediator services among*

Table 4. Template-based Specification of Change Pattern Instance

Name = Corelated Inclusion, Id = 3, CLS = 1, Intent =“...”, Frequency = 8			
Precondition(s): as in Figure 4a, PostCondition(s): as in Figure 4c			
cID	**OPR**	**Architecture Elements**	**Parameters**
1454	ADD()	CustomerAccount ∈ CMP	“ ”
1455	ADD()	CustomerDebt ∈ CMP	“ ”
1456	ADD()	recurringPayment ∈ CON	CustomerAccount, CustomerDebt ∈ CMP
1457	ADD()	billAmount ∈ CON	Biller_CRM, CustomerAccount ∈ CMP
1458	ADD()	paidAmount ∈ CON	Biller_CRM, CustomerDebt ∈ CMP
1459	REM()	customerTariff ∈ CON	Biller_CRM, CustomerPayment ∈ CMP
1460	REM()	paymentInvoice ∈ CON	Biller_CRM, CustomerInvoicing ∈ CMP
1461	ADD()	makePayment ∈ CON	CustomerAccount, CustomerPayment ∈ CMP
1462	ADD()	getReceipt ∈ CON	CustomerDebt, CustomerInvoicing ∈ CMP

two or more directly connected service components. The column cID represents the sequences of change as it is captured in the change log and later as individual graph nodes. Figure 4 represents a partial architecture view for the EBPP case study (integrate direct debit to customer accounts and adjust customer debt) captured as a recurring sequence ($Freq(S_C) = 8$) in Table 4.

For example, in Figure 4 the preconditions specify the components (*Biller_CRM, CustomerPayment, CustomerInvoicing*) and connectors (*customerTariff, paymentInvoice*) must exist in the architecture before a pattern can be applied. In addition, the post-conditions specify the addition of new components (*CustomerAccount, CustomerDebt*) and connectors (*makePayment, recurringPayment, getReceipt*) as a result of pattern-driven change execution. The change operations specify the execution aspects in terms of addition or removal of specified elements from the architecture, illustrated in Figure 4b.

5.2 Algorithmic Analysis

Pattern identification from change logs, which can potentially be significant in size, requires an efficient solution. In our trials, we observed that the preprocessing for a significant graph size (i.e, $G_C.size() = OPR \geq 2278$) remains constant with average complexity time = 888.9 ms. However, such pre-processing is fundamental to our approach and the benefit for candidate validation lies in eliminating the potential patterns (false positives) that may violate the structural integrity of an architecture. We customise the input parameters as: $minLen(S_C) = 2, maxLen(S_C) = 9$ with total change operations: $G_C.size() = 2278$. In addition, we increase the pattern frequency threshold $Freq(S_C)$ by 2 in each trial, where $Time \propto Freq(S_C)$ and $Freq(S_C) \propto 1/Instances$. The technical difference between the exact and inexact pattern matching is detailed in Section 4. The summary of comparison (on average): time (exact : inexact) in milliseconds = T(564:1214) ms and identified patterns instances (exact : inexact) = I(21:38), for $G_C.size() = 2278$.

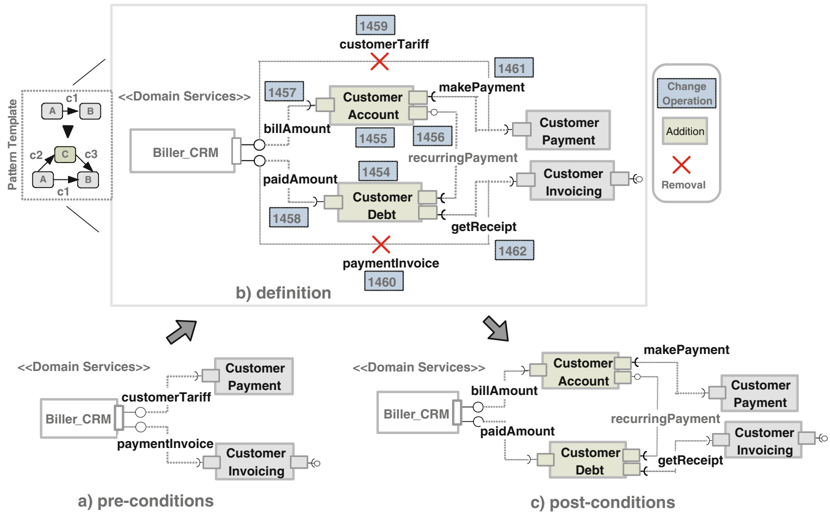


Fig. 4. An Example of the Identified (Co-related Change) Pattern Instance

Possible Limitations: The proposed approach falls short of capturing dynamic dependencies in terms of service compositions that correspond to the behavioral aspects in SOAs. These dynamic dependencies go beyond structural graph matching and is out of the scope for this research. The limitation is inherent in the change log that only captures association type connectors that correspond to structural changes. In addition, change patterns do not necessarily support an optimal solution to architecture evolution problem; instead they promote an alternative and potentially reusable solution.

6 Related Work

Two areas - pattern-driven change reuse and graph-based pattern identification - play a role in our research. The solutions for *pattern-based architecture evolution* utilise evolution styles [5] and more specifically "evolution shelf" [13] as a generic infrastructure to achieve for-reuse and by-reuse techniques for software architecture evolution. It aims at supporting refactoring patterns (i.e., add a component, move a component etc.) that can be composed into further advanced evolution styles (add a client, move a client etc.). In contrast to the evolution styles [5, 13] for more conventional component architectures, we observe that operationalisation of changes in the log exhibits process-centric patterns of change unlike the frequent addition or removal of individual components and connectors.

A catalog of process change patterns [15] can guide change in process-aware information systems. In contrast, we exclusively focus on change operationalisation for architectural abstraction. This allows us to argue about change patterns as generic, first class abstraction that can be specified once and instantiated multiples times to support potential reuse in architecture-based change execution. We follow ideas in [16] that utilise process change logs to gain an empirical insight into the context and scope for process instance changes. Our solution focuses on fostering the common architectural changes that could guide the architects to follow a reuse-centered approach for architectural change execution.

The solution to our pattern identification problem is similar to other *graph-based pattern identification* techniques based on frequent sub-graph mining techniques [4]. We use an apriori-based approach with Breadth First Search strategy for iterative graph matching. In this context, Graph X-Ray (G-Ray) [10] works on attributed graphs to find subgraphs that either match the desirable query pattern exactly, or as close as possible based on pre-defined criteria. We are specifically concerned with identifying patterns in medium to large attributed graphs where graph nodes and edges may have multiple attributes that contain instances of architecture elements and pattern-specific constraints.

7 Conclusions

Service software evolves as a consequence of business and technical change cycles. Scalability beyond manual evolution and change support is required to enable a systematic change reuse during architecture evolution. Investigating the history of sequential architectural changes allows post-mortem analysis to identify patterns as generic, potentially reusable solution for software architecture evolution. The contribution of this paper is a graph-based formalism for architecture change representation that allows automation along with parameterised customisation to identify change patterns.

In the future, we will focus on developing a pattern catalogue as a repository infrastructure to support an automated storage and retrieval of change patterns. We utilise an XML pattern template that allows for once-off abstract specification for identified patterns [17] that can be queried and instantiated with concrete pattern instances to support potentially reusable architecture evolution.

References

1. Buckley, J., Mens, T., Zenger, M., Rashid, A., Kniesel, G.: Towards a Taxonomy of Software Change. Jrnl of Software Maintenance and Evolution 17, 309–332 (2005)
2. Ahmad, A., Pahl, C.: Pat-Evol: Pattern-drive Reuse in Architecture-based Evolution for Service Software. ERCIM News 88 (January 2012)
3. Ahmad, A., Pahl, C.: Customisable Transformation-Driven Evolution for Service Architectures. In: Europ. Conf. on Software Maintenance and Reengineering, CSMR 2011. Doct. Consort. (2011)
4. Jiang, C., Coenen, F., Zito, M.: A Survey of Frequent Subgraph Mining Algorithms (2004)
5. Garlan, D., Barnes, J., Schmerl, B., Celiku, O.: Evolution Styles: Foundations and Tool Support for Software Architecture Evolution. In: Proceedings of the Joint Working IEEE/IFIP Conference on Software Architecture (2009)
6. Gruhn, V., Pahl, C., Wever, M.: Data Model Evolution as Basis of Business Process Management. In: 14th International Conference on Object-Oriented and Entity Relationship Modelling O-O ER 1995. LNCS Series. Springer (1995)

7. Javed, M., Abgaz, Y.M., Pahl, C.: A Pattern-Based Framework of Change Operators for Ontology Evolution. In: Meersman, R., Herrero, P., Dillon, T. (eds.) OTM 2009 Workshops. LNCS, vol. 5872, pp. 544–553. Springer, Heidelberg (2009)
8. Ehrig, H., Prange, U., Taentzer, G.: Fundamental Theory for Typed Attributed Graph Transformation. In: Ehrig, H., Engels, G., Parisi-Presicce, F., Rozenberg, G. (eds.) ICGT 2004. LNCS, vol. 3256, pp. 161–177. Springer, Heidelberg (2004)
9. Pahl, C.: A Formal Composition and Interaction Model for a Web Component Platform. In: ICALP 2002 Workshop on Formal Methods and Component Interaction. ENTCS (2002)
10. Tong, H., Faloutsos, C., Gallagher, B., Eliassi-Rad, T.: Fast Best-Effort Pattern Matching in Large Attributed Graphs. In: 13th ACM International Conference on Knowledge Discovery and Data Mining, KDD 2007, pp. 737–746 (2007)
11. Fahmy, H., Holt, R.C.: Using Graph Rewriting to Specify Software Architectural Transformations. In: 15th Intl. Conf. on Automated Software Engineering (2000)
12. Lewis, G., Smith, D.B., Kontogiannis, K.: A Research Agenda for Service-Oriented Architecture (SOA): Maintenance and Evolution of Service-Oriented Systems. Technical report, Software Engineering Institute (2010)
13. Goaer, O.L., Tamzalit, D., Oussalah, M., Seriai, A.D.: Evolution Shelf: Reusing Evolution Expertise within Component-Based Software Architectures. In: 32nd Annual IEEE Intl. Computer Software and Applications Conference (2008)
14. Ng, R., Lakshmanan, L., Han, J., Pang, A.: Exploratory Mining and Pruning Optimizations of Constrained Associations Rules. In: SIGMOD 1998 Conference (1998)
15. Weber, B., Rinderle, S., Reichert, M.: Change Patterns and Change Support Features in Process-Aware Information Systems. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007 and WES 2007. LNCS, vol. 4495, pp. 574–588. Springer, Heidelberg (2007)
16. Günther, C.W., Rinderle, S., Reichert, M., van der Aalst, W.: Change Mining in Adaptive Process Management Systems. In: Meersman, R., Tari, Z. (eds.) OTM 2006, Part I. LNCS, vol. 4275, pp. 309–326. Springer, Heidelberg (2006)
17. Barrett, R., Patcas, L.M., Murphy, J., Pahl, C.: Model Driven Distribution Pattern Design for Dynamic Web Service Compositions. In: International Conference on Web Engineering, ICWE 2006. ACM Press (2006)

Multi-Tenancy Multi-Target (MT^2): A SaaS Architecture for the Cloud

Antonio Rico Ortega[1], Manuel Noguera[1], José Luis Garrido[1], Kawtar Benghazi[1], and Lawrence Chung[2]

[1] Departamento de Lenguajes y Sistemas Informáticos, Universidad de Granada, E.T.S.I.I.T., c/ Periodista Daniel Saucedo Aranda s/n, 18071 Granada, Spain
{antoniorico,mnoguera,jgarrido,benghazi}@ugr.es
[2] Department of Computer Science, University of Texas at Dallas, Richardson, Texas 75083, USA
{chung}@utdallas.edu

Abstract. Multi-tenancy (MT) architectures allow multiple customers to be consolidated into the same operational system. Multi-tenancy is key to the success of Software as a Service (SaaS) by means of a new software distribution formula in which customers share application and costs are indirectly assumed by all of them. However, as traditional applications do, each MT application deploys a single functionality, therefore component sharing between applications only occurs in an ad hoc manner and thereby hindering software reuse. In this paper it is introduced Multi-tenancy Multi-target (MT^2), an extension to MT Architectures for the development and deployment of one single software application encompassing several functionalities. To this end, some new components are added to traditional MT Architectures, thus providing new benefits for software developers, vendors and clients, and which are described by means of real examples.

Keywords: multi-tenancy, cloud computing, software as a service, software architecture.

1 Introduction

Cloud Computing has brought high computational resources to everyone [1], so that small and medium-size software vendors have now the opportunity to access high processing capabilities so far reserved to big corporations. Vendors develop new applications offered through Internet as a service, while customers access them through web browsers anywhere and anytime. This new model of software distribution is called Software as a Service (SaaS) [2]; clients subscribe to vendor's application services and pay for using them [2–4]. Customers afford top-software deployments eliminating initial investment and operational expenses.

Multi-tenancy is becoming a key technology for the success of SaaS since clients reduce the cost of software use by sharing expenditures, whereas software vendors maximize sales profits. Multi-tenancy Architectures (MTA) allow multiple customers

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 214–227, 2012.

(aka *tenants*) to be aggregated into the same application. Tenants share not only application, but also capital and operational expenses [5]. Moreover, tenants are also able to customize their applications both in endpoint presentation and data structure according to their particular needs.

In this context, demand has to be supported by an MTA that allows agile accounts creation in the system. Basically, MTA models have two tiers: administrative and instance; the administrative tier [5] provides the functionalities responsible for creating and managing tenants accounts, while the instance tier hosts the applications that tenants execute according to subscription contracts defined at the administrative level.

Traditional multi-tenant applications are shared among tenants with common functional needs. However, each MT application usually deploys one single functionality and therefore component sharing between applications only occurs in an ad hoc manner at lower levels in the architecture and basic shared components need to be replicated for each application.

In this paper we introduce a proposal called Multi-tenancy Multi-target (MT^2), as an extension to multi-tenancy architecture (from now on we will also refer to it as *mono-target architecture*). MT^2 allows multiple functionalities to be offered in the same operational system. This way, applications are distributed among tenants with different functional needs and vendors can host tenants from heterogeneous market sectors. This multifunctional situation seeks for several benefits: companies are able to subscribe to only one SaaS application; vendors have a multi-target market, broadening the spectrum of potential customers; and developers reach agility by avoiding unnecessary replications.

The rest of the paper is organized as follows. Section 2 provides a background on Cloud Computing, SaaS, and Multi-tenancy technologies. The MT^2 proposal is introduced in Section 3. The architectural design decisions supporting the proposal, and also a real development based on it (called Globalgest), are described in detail in Section 4 and 5, respectively. Next, a Section 6 discusses the relevance of the contribution and related work, and finally Section 7 summarizes conclusions and future work.

2 Cloud Computing Technologies

Cloud computing, and related technologies such as SaaS and Multi-tenancy, are producing a big change in comparison with traditional models for software distribution and use. There is still no common agreement about the definition of *Cloud Computing*, and actually, some authors use Cloud Computing as a synonym of *Utility Computing* [6]: "*A computing Cloud is a set of network enabled services, providing scalable, QoS guaranteed, normally personalized, inexpensive computing platforms on demand, which could be accessed in a simple and pervasive way*".

In this paper, we will use Berkeley's definition by which Cloud Computing is defined as the sum of Utility Computing and Software as a Service [7]. Utility Computing [8] refers to the use of computer resources on demand and it enables a distribution formula for software vendors called *Software as a Service* (SaaS). According to [2] *"The basic long-term vision of SaaS is centred around separating software possession and ownership from its use"*. Unlike its predecessor *Software on Premises*, applications are now installed in a Cloud and accessed over Internet; users are not owners of the software, but consumers of web applications.

Figure 1 shows actors of Cloud Computing. The datacenter (Cloud Provider) serves utility computing to Cloud Users who provide applications on-demand (Software as a Service) to tenants (SaaS users).

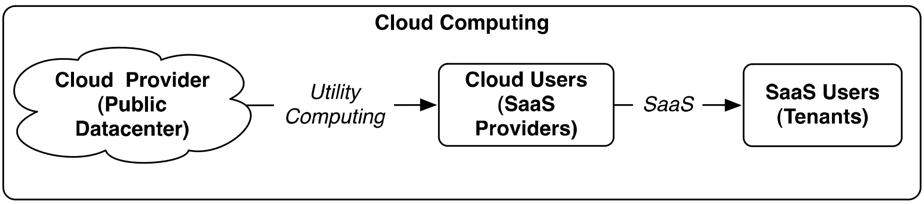


Fig. 1. Actors in Cloud Computing (based on **[1]**)

2.1 Multi-tenancy

SaaS is not the first distribution formula based on outsourcing and software access through wide area networks. However, it is the first in succeeding unlike previous similar attempts like *Application Service Provider* (ASP). Among other reasons, ASP failed because it did not even contemplate the possibility to serve different companies using the same software instance [9] or the ability to provide customized applications [10].

Multi-tenancy is a software architecture that leverages economy of scale by the aggregation of users (tenants) into the same application; software instance is shared among tenants, and so are expenditures.

Figure 2 illustrates multi-tenant system architecture. The lower level tiers perform changes dictated by business layer in both database and file system. Intermediate layers such as presentation or SOA services communicate with browser and smart devices respectively to produce end-users output. Metadata are responsible for system customization so that tenants can get a specific user experience. This customization includes data model extension, adaptation of presentation layer to corporative image and business workflow personalization. Security services must be present in all multi-user systems. In multi-tenant environments, the complexity of this component increases; systems must maintain privacy not only among end-users, but also among different tenants.

Customization and security relay on the model chosen to store data. Several authors have proposed different approaches ([11–16]) of database models in multi-tenancy; though with different terminology, they all agree that the distinction is given by the level of isolation on tenants data [17]. Dean Jacobs, on its article "Ruminations on multi-tenant databases" [5] suggests three approaches:

- Shared Machine: High degree of isolation
- Shared Database: Medium degree of isolation
- Shared Tables: Lower degree of isolation

Regarding the isolation of the database layer, the more isolated the data is among different tenants, the easier to customize, but the more expensive are hardware and maintenance. According to [18], multi-tenancy is pure when using low degrees of isolation (like in the Shared Tables approach); other variations where reutilization of resources is not maximized are considered as semi-multitenant.

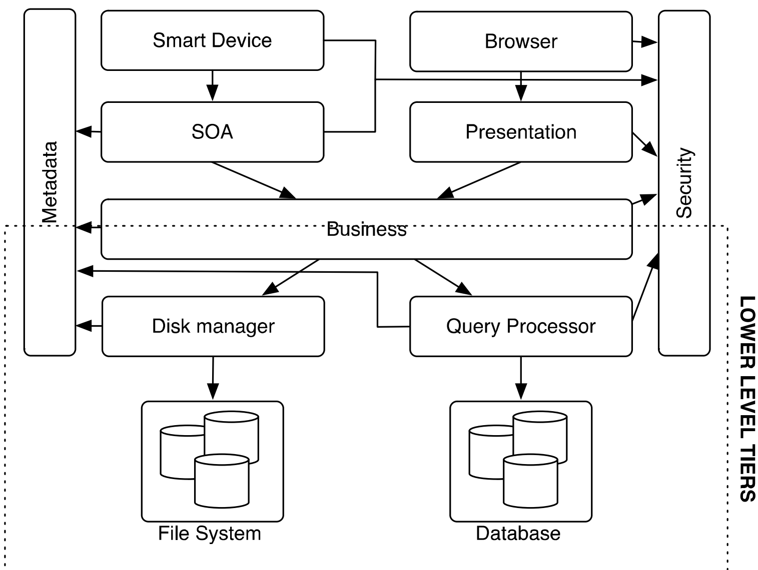


Fig. 2. MT General Architecture model

3 An Approach Extending Multi-tenancy Architecture: MT^2

SaaS applications are highly scalable due to Multi-tenant efficiency [16][17]. Expenses are defrayed among all customers sharing the same software and operational system. As well as instances, users share application functionality. However, current multi-tenancy applications deploy just one single functionality or are aimed to serve a specific line-of-business (*LOB*). With this model, companies have to subscribe to as many applications as services they need. For instance, a company needing *Customer Relationship Management* (CRM), *Content Management System* (CMS) and *Enterprise Resource Planning* (ERP) functionalities would have to contract a different subscription for each of them (see Tenant 2 in Figure 3-a).

In this *mono-target* situation, vendors will have the potential clients spectrum limited by the functionality their applications deploy. For example, CMS vendors will focus on companies needing CMS solutions, whereas CRM vendors will target companies looking for solutions for customer relationship problems.

Traditional multi-tenancy divides the set of potential clients into highly disjoint sets. Companies targeting one set will have to develop new applications if they want to reach other sets of the market. Figure 3-a shows how tenants with different needs subscribe to different applications, since no application can deploy several functionalities (Tenant 2 has subscriptions to three different applications).

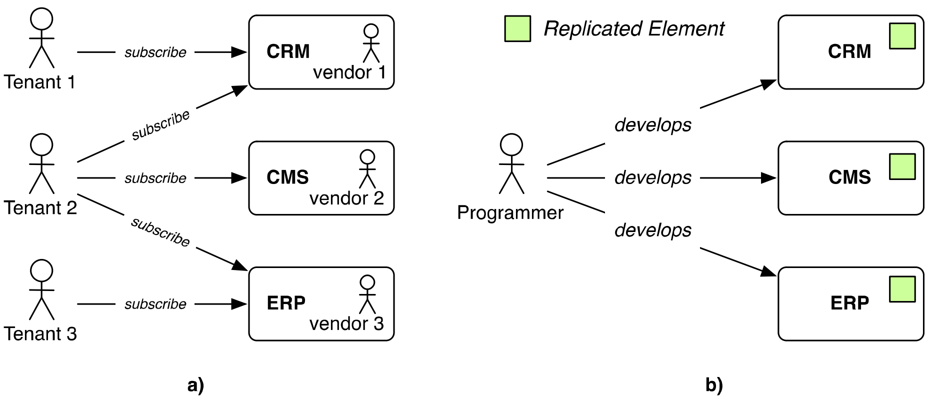


Fig. 3. a) Subscriptions depend on functional needs b) Replication of common development components

At development level, basic features such as user authentication and database connection are common and can be shared in CRM, CMS or ERP applications. Since different software functionalities are hosted in different implementations, these lower level components are to be replicated along all implementations (see Figure 3-b). This replication increases programmers' effort and therefore time-to-market.

3.1 MT^2 Foundations

The main idea behind the MT^2 approach is to allow multi-tenancy systems to deploy not only one single functionality, but several ones. MT^2 extends traditional Multi-tenancy so that tenants with different functional needs could be able to make use of customized end-user applications while sharing the same underlying software system.

The set of functionalities deployed in a MT^2 system is called *functional portfolio*. The number of functionalities in the portfolio may differ depending on vendor. Tenant subscriptions are defined by a subset of functionalities within the same functional portfolio.

MT^2 systems seek for scalability not only at the tenant level, but also at functional level. Young MT^2 systems may deploy just a few features, but can increase portfolio across the time. Old MT^2 are supposed to have larger functional portfolios, since new functionalities are added on customers' demands and remain on the portfolio, unless outdated.

In this way, companies are able to have multiple functionalities through just one software application. With MT^2, tenants could use just one application to serve all their functional needs. In the last example, the company (Tenant 2) will now be able to unify all its functional needs in just one MT^2 application (see Figure 4-a).

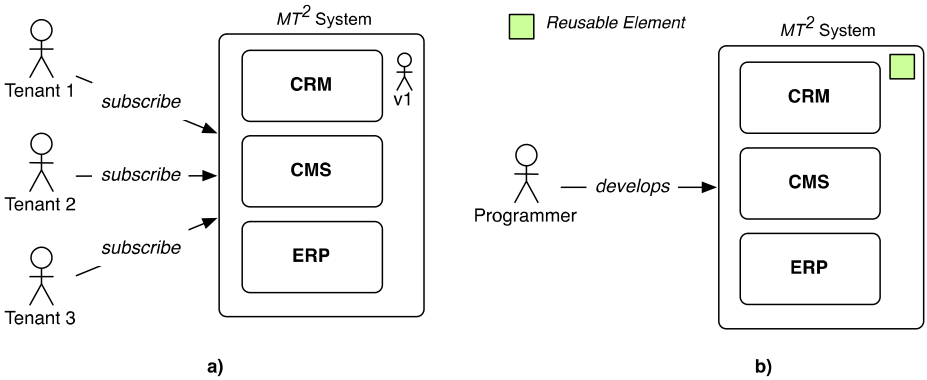


Fig. 4. a) MT^2 systems allow tenants to subscribe to multiple functionalities and broaden the spectrum of potential clients b) Reusability of common resources in MT^2

MT^2 also seeks for to increase the range of potential customers. The perfect separation of potential clients in traditional multi-tenancy disappears and vendors reach a wider range of targets by mixing and overlapping disjoint sets. Figure 4-a shows how a single vendor (v1) can access different markets by offering different functionalities. The larger the portfolio is, the bigger the number of potential customers and opportunities is.

New features involve new sales opportunities. When vendors decide to increase functional portfolio adding *Document Management System* (DMS) functionalities, for instance, chances to raise profits increase. Besides for the demanding tenant, this new development could be available for the rest of tenants (upgrading subscription) and the rest of potential clients in the market.

MT^2 pays special attention to achieve reusability by removing useless replication of common features. In multi-tenancy mono-target, functionalities are deployed in different applications; hence components are replicated. In multi-target, shared components are reutilized among all functionalities reducing time-to-market and development effort. Figure 4-b shows how MT^2 changes replication for resource reutilization.

4 General Model of MT^2 Architecture

Extra components are added to traditional multi-tenant architectures so as to provide a (now) multi-functional subscription. These modifications are present both at administrative and instance level. Figure 5 shows how MT^2A includes those new components (marked in red) on the basis of the MT architecture.

MT² is independent of the underlying multi-tenant architecture. Any MT system could be upgraded to MT² regardless of design aspects such as the isolation degree in the database. Multi-target extension is based on the reusability of lower level components during development process; many basic features are shared along applications different in nature. In MT² all these components (libraries, functions, icons, graphics, style sheets, etc.) are no longer replicated, but reused. Modifications to the traditional MT architecture are mainly focused on:

- Security commitments
- Multi-target metadata for contracts
- Business process reutilization

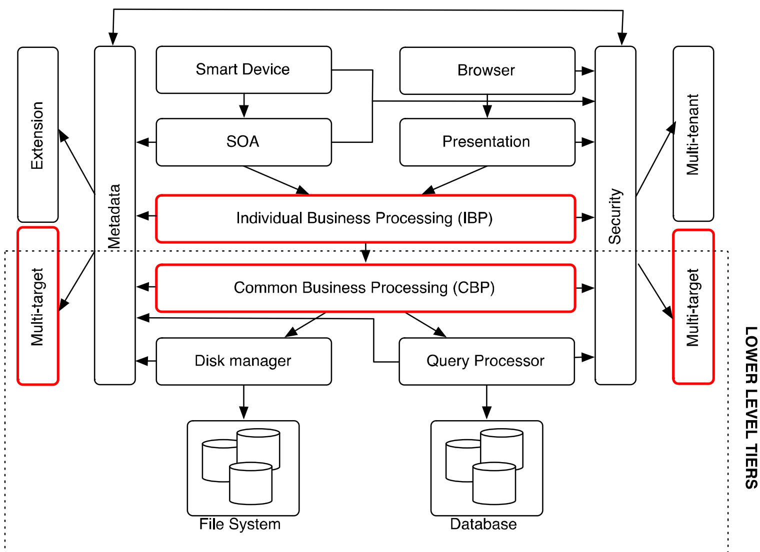


Fig. 5. MT2 Architecture Model

4.1 Security Commitments

Multi-target applications deploy different functionalities depending on tenant's subscription; tenants share application, but functional deployment may differ. In this situation, security components become more complex in architecture, since end-users are allowed to execute those functionalities present in the subscription and not others. Multi-target involves new commitments for security layer at two levels:

- *Tenant level*: Tenants should not deploy functionalities that are not included in subscription. Security must ensure that forbidden functionalities are not deployed.
- *End-user level*: Multi-tenancy applications are multi-user environments at instance level. Tenants end-users have different roles that determine their

capabilities in the system. In MT^2, tenants may have subscription to one functionality, but not all tenants end-users should have access to it. Admin users of the tenant must have the capability to decide for each user what functionalities deploy from the tenant portfolio.

4.2 Multi-target Metadata for Contracts

Multi-target metadata links tenants accounts to functionalities controlling not only functionalities subscribed by tenants, but also contractual features of this relation. For instance, if a tenant wants to subscribe to *SMS* functionality, at least we should set the number of text messages contracted; by setting this parameter in other functionalities such as *Client Management*, does not make sense. Every subscription to functionalities has its own conditions and these are reflected on the Multi-target metadata of each tenant. As well as subscription terms, this component is responsible for:

- Synchronizing with security services and inform about subscription details to prevent access to forbidden functionalities.
- Determining which *individual business processes* to import on *Individual Business Processing* (IBP) layer.

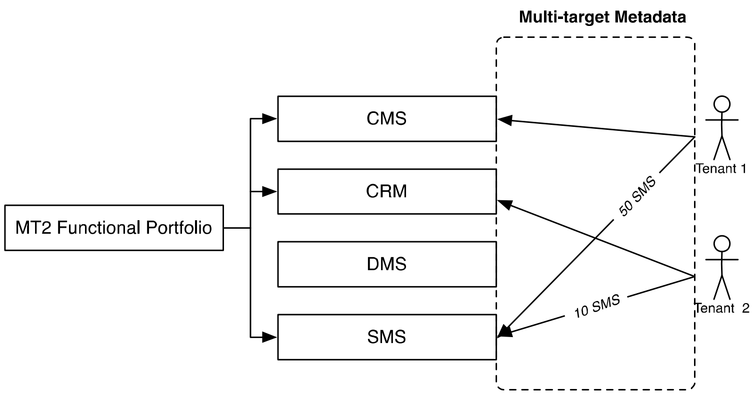


Fig. 6. Multi-target metadata contains subscription details to functionalities

Previous figure shows an example of an MT^2 system with 4 functionalities in the portfolio and two tenants. Subscriptions are defined by Multi-target metadata; both tenants have contracted SMS, but the number of text messages to be sent differs.

4.3 Business Process Reutilization

Reusability of common features along all functionalities is the main cause of this MT^2 extension. In a multi-target environment, business layer is divided into two:

- *Common Business Processing (CBP)*: It contains those elements business-independent and reusable across all functionalities
- *Individual Business Processing (IBP)*: It includes those elements that are business-dependent and which are specifically designed to support one functionality.

During the execution timeline of their application instances, all tenants will import CBP elements statically; however IBP elements will be imported dynamically depending on tenant's subscription. In Figure 7, tenant has a subscription to functionalities F2 and F4. All CBP are imported statically, but just F2 and F4 IBP elements will be imported, since these two functionalities will be deployed in the execution.

CBP represents all those components reused in different functionalities. Features such as privacy or system authentication are no longer to be developed in future functionalities; they are already in the CBP layer. Furthermore, development effort is reduced not only because of reutilization of CBP elements, but also for the extension of them. For instance, if a programmer needs to develop a specific feature and encapsulate it within a class, that class does not need to be developed from the beginning, but it can be coded by extending one existing class from the CBP layer.

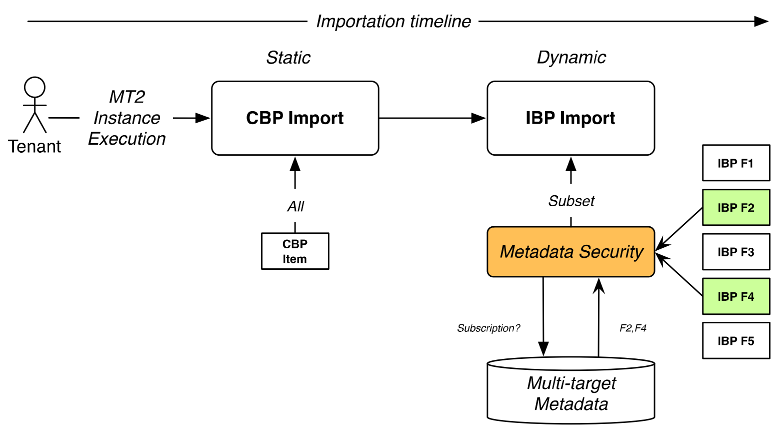


Fig. 7. Static and Dynamic import in MT^2

5 Globalgest, a Software with MT^2 Architecture

Globalgest [19][20] is an example of a business-oriented application based on MT^2 architecture. Installed in 17 companies, Globalgest deploys more than 100 functionalities. Combinations of this portfolio allow Globalgest to serve businesses from different industries such as a medical clinic or an IT company (see Figure 8 and Figure 9). As we see on both figures, tenants share application instance, but do not need to share functionalities or presentation. Globalgest allow tenants to customize application interface by using personalized style sheets and graphics.

Functional portfolio in Globalgest increases on customer demands. Whenever a tenant needs a new functionality, this is developed and included in service portfolio. Existing tenants could subscribe to the new feature and *vendor leads (i.e. potential customers)* could be converted due to this functional improvement. For instance, last functionality developed has been e-commerce connection; this feature (developed ad hoc for one specific tenant) is now available for other tenants who might upgrade their subscription to incorporate on-line selling to their websites.

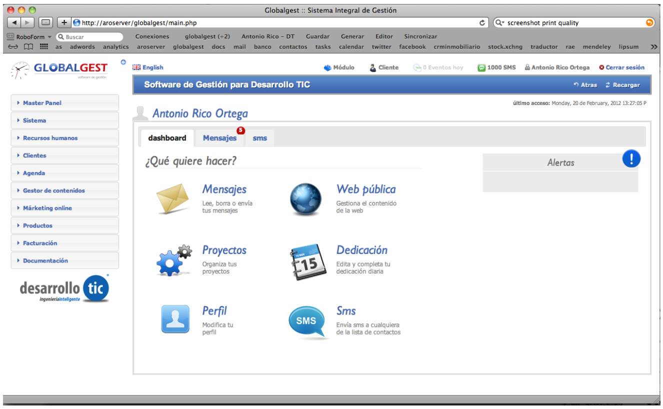


Fig. 8. Globalgest: IT Company Implementation

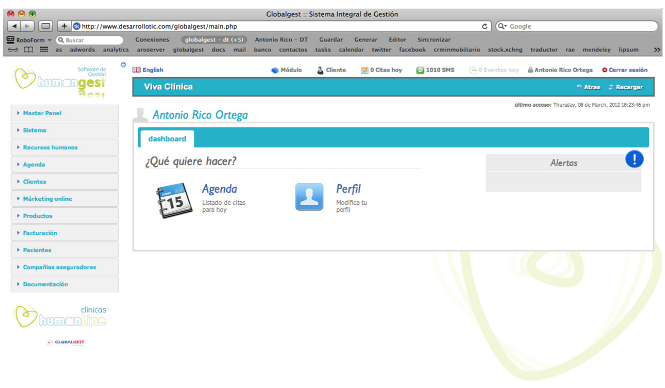


Fig. 9. Globalgest: Medical Clinic implementation

Figure 10 illustrates a diagram with some of the functionalities present in Globalgest portfolio. Green boxes represent the subscription of a medical center, blue boxes are those contracted by an IT company and white boxes represent common subscriptions of them both. In this case, medical's center functionalities were programmed ad hoc for the client, but once developed they remain in the functional portfolio of Globalgest. This means that if another medical company requires them, now these functionalities will be available. This new medical company may not need invoicing, but will likely be interested in patient management, insurance companies monitoring and/or sending programmed SMS reminders for appointments.

Globalgest is real MT^2 software that proves how a single application serves two companies from different industries without duplicating the effort. MT^2 architecture allows Globalgest to deploy and host several functionalities configuring client functional subscription on demand.

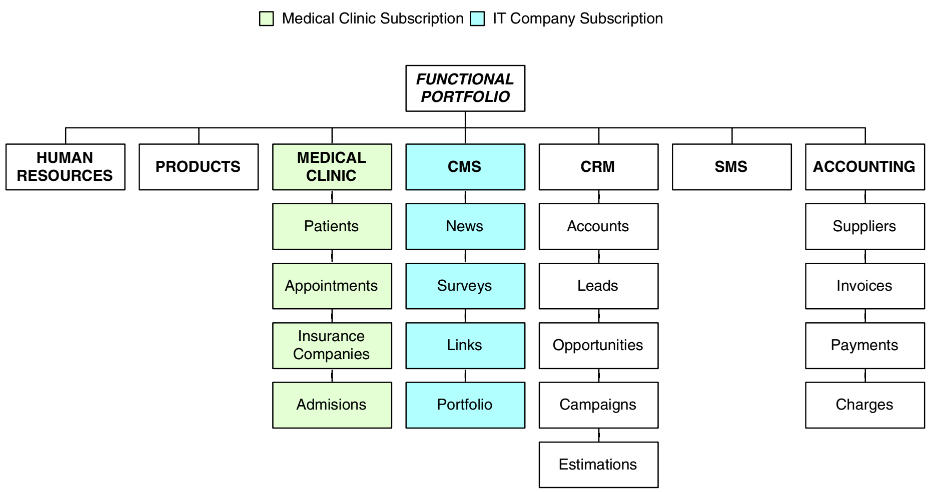


Fig. 10. Medical clinic and IT Company subscriptions

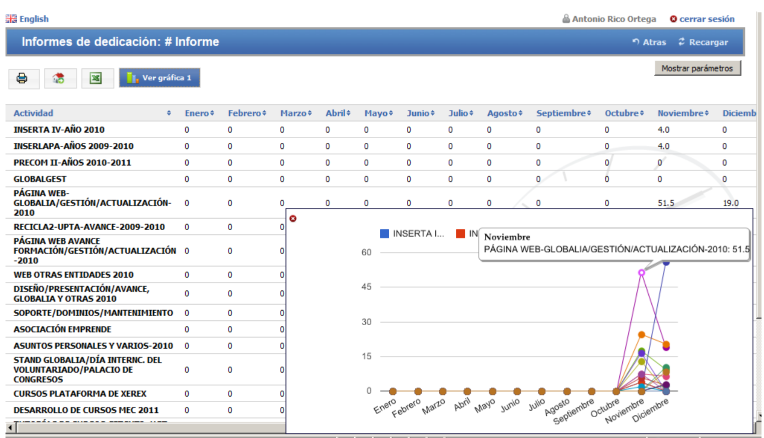


Fig. 11. Worker performance report in Globalgest

As well as features showed in Figure 10, other functionalities implemented in Globalgest and already serving companies are:

- E-Commerce with order management
- Presence Control.
- Workers performance management and reports (Figure 11)
- Financial management and reports
- Teaching companies management: courses, registrations, students, teachers and tutors, classrooms availability, etc.
- Foundation companies management: subsidies, volunteers, human projects.

6 Discussion and Related Work

The market of SaaS Multi-tenancy applications is broad. Clients pay for the use of many different types of software deploying different functionalities such as ERP, CRM, CMS, DMS, etc. Each SaaS application offers a specific functionality to its customers. Clients subscribe to vendor services in a pay-per-use basis. This single-functionality deployment has some inconveniences, not only for customers, but also for software vendors and developers.

When a company needs a particular functionality such as that of a CRM, it compares among all vendors and chooses the one that best fits both its budget and needs. If this same company needs another functionality, e.g, that of a CMS, it will have to repeat the same process by subscribing to another different software application from the same or a different provider. In this case, companies contract as many software subscriptions as functionalities they need. So long as prices increase, users will have to learn the use of several different software interfaces. In this 'mono-functional' context, the spectrum of vendors' potential clients is limited by the purpose each application serves. For example, a CMS vendor will focus on companies needing CMS solutions and not needing other functionalities.

From a developer point of view, there are many basic components that could be shared in applications of different nature. In lower level development, features such as user authentication and database connection are common, regardless of the kind of application (CRM, CMS or DMS). However, these lower level components are to be replicated along all implementations, since different software functionalities are hosted in different application instances.

Software Product Line Approach (SPLA) is also based on the reusability of elements called *artifacts* for delivering affordable customized applications. However, while reusability takes place during development process in SPLA, MT^2 reuses CBP components during execution. Furthermore, we consider in a higher abstraction level of meta applications than that of SPLA mass production. In MT^2, customers not only get to customize their instance, but they can also have a completely different application instantly by changing their subscriptions. Both approaches pursue similar goals and benefits such as reductions of expenses and time-to-market, but exploiting different means.

Multi-tenancy is a novel paradigm. Proposals and implementations are scarce and studies for this area are mainly schema-mapping techniques and benchmarking ([11–17]). Based on the elimination of useless replications, this paper takes a different approach and proposes modifications to the general AMT; MT^2 is an extension applied to the whole architecture rather than just the database layer. This modifications work towards obtaining a new kind of SaaS applications with benefits for all parties involved in the industry of software.

7 Conclusions

Multi-tenancy permits several customers to share network applications in the *Cloud*. This is essential for the success of SaaS as software distribution formula; users are no longer owners of software but tenants of it. In this model, operational and maintenance costs are also allocated along users.

Traditionally, applications aim to serve one single purpose. For instance, CRMs are used for managing customers' accounts and *leads* conversion, but not for other functionalities. Therefore, different purposes involve different applications. There are many basic components that could be reused along all these different implementations, like authentication, payrolls, message broadcasting and notification, etc.; however, developers do replicate it along all implementations. Commercially speaking, traditional Multi-tenancy can be called Mono-target: vendors have their potential clients limited by software purposes. The users themselves have to subscribe to as many services as they need; hence incrementing costs and increasing learning effort.

This paper has presented the MT^2 architecture that is supported by an extension to the current Multi-tenant architectures. MT^2 tries to go a step further in relation to the current MT applications. MT^2 turns traditional Multi-tenancy software into Multi-target applications. MT^2 is based on the elimination of unnecessary replication and the reutilization of lower level components of the architecture. MT^2 applications deploy several functionalities and tenants choose which one to subscribe to. Users unify all applications in one and thus, reduce expenses and effort; vendors broaden the range of potential clients and developers speed up implementation and development.

In order to illustrate the applicability of the proposal, we have introduced Globalgest, a commercial software based on the MT^2 architecture that shows how one individual application can host tenants from different lines of business.

Acknowledgements. This work has been partially funded by CEI-BioTIC Granada under project 20F2/36 and the Andalusian Regional Government under projects SEJ-6765 and TIC-6600.

References

[1] Armbrust, M., Fox, A., Griffith, R., Joseph, A.: A view of cloud computing. Communication of the ACM (2010)

[2] Turner, M., Budgen, D., Brereton, P.: Turning software into a service. Computer 36(10), 38–44 (2003)

[3] Motahari-nezhad, H.R., Stephenson, B., Singhal, S.: Outsourcing Business to Cloud Computing Services: Opportunities and Challenges. Development (2009)

[4] Dubey, A., Wagle, D.: Delivering software as a service. McKinsey Quarterly (May 2007)
[5] Jacobs, D., Aulbach, S.: Ruminations on multi-tenant databases. In: Fachtagung für Datenbanksysteme in Business, Technologie und Web, Aachen, Germany, pp. 5–9 (March 2007)
[6] Wang, L., Tao, J., Kunze, M., Castellanos, A.C., Kramer, D., Karl, W.: Scientific cloud computing: Early definition and experience. In: 10th IEEE International Conference on High Performance Computing and Communications, HPCC 2008, pp. 825–830 (2008)
[7] Armbrust, M., et al.: Above the clouds: A berkeley view of cloud computing. EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28 (2009)
[8] Parkhill, D.F.: Challenge of the Computer Utility, p. 232. Addison-Wesley Educational Publishers Inc. (1966)
[9] Liu, G., Jiang, H., Geng, R.: Software design on a SaaS platform. In: 2010 2nd International Conference on Computer Engineering and Technology, pp. V4-355–V4-358 (2010)
[10] Papazoglou, M.P.: Service -Oriented Computing: Concepts, Characteristics and Directions. Information Systems
[11] Aulbach, S., Jacobs, D., Kemper, A.: Multi-Tenant Databases for Software as a Service: Schema-Mapping Techniques. Techniques, 1195–1206 (2008)
[12] Aulbach, S., Seibold, M., Jacobs, D., Kemper, A.: A Comparison of Flexible Schemas for Software as a Service. Acme, 881–888 (2009)
[13] Chang, F., Dean, J., Ghemawat, S.: Bigtable: A distributed storage system for structured data. ACM Transactions on (2008)
[14] Copeland, G.P., Khoshafian, S.N.: A decomposition storage model. ACM SIGMOD Record 14(4), 268–279 (1985)
[15] HBase Storage Architecture (2009), http://www.larsgeorge.com/2009/10/hbase-architecture-101-storage.html
[16] Chong, F., Carraro, G.: Architecture Strategies for Catching the Long Tail What is Software as a Service? Most 479069, 1–22 (2006)
[17] Chong, F., Carraro, G., Wolter, R., Corporation, M., Architecture, A.: Multi-Tenant Data Architecture Three Approaches to Managing Multi-Tenant Data. Architecture 479086, 1–18 (2006)
[18] Bezemer, C.-P., Zaidman, A.: Challenges of Reengineering into Multi-Tenant SaaS Applications. Challenges (2010)
[19] Rico, A.: Globalgest SaaS - Software as a Service, http://globalgest.info/ (accessed November 01, 2011)
[20] Rico, A.: Desarrollo TIC. SEO, Web, and Software Development, http://www.desarrollotic.com/ (accessed February 15, 2012)

An Architecture for Efficient Web Crawling*

Inma Hernández, Carlos R. Rivero, David Ruiz, and Rafael Corchuelo

University of Sevilla, Spain
{inmahernandez,carlosrivero,druiz,corchu}@us.es

Abstract. Virtual Integration systems require a crawling tool able to navigate and reach relevant pages in the Deep Web in an efficient way. Existing proposals in the crawling area fulfill some of these requirements, but most of them need to download pages in order to classify them as relevant or not. We propose a crawler supported by a web page classifier that uses solely a page URL to determine page relevance. Such a crawler is able to choose in each step only the URLs that lead to relevant pages, and therefore reduces the number of unnecessary pages downloaded, minimising bandwidth and making it efficient and suitable for virtual integration systems.

Keywords: Web Crawling, Crawler Architecture, Virtual Integration.

1 Introduction

Virtual Integration aims at accessing web information in an automated manner, considering the Web as a source of information. The Virtual Integration process starts with a query, in which the user expresses her interests, and its goal is to obtain information relevant to that query from different sites of the Web (usually Deep Web sites), and present it uniformly to the user in a transparent way. From now on, we refer to relevant pages as web pages containing information related to the user interests. Note that these interests may change over time, so the same page can be either relevant or irrelevant.

This process is online, which means that while the system is looking for information, the user is waiting for a response. Therefore, bandwidth and efficiency are important issues for any Virtual Integration approach, and downloading only relevant pages is mandatory.

Virtual Integration processes require a tool able to navigate through web sites, looking for the information. A web crawler is an automated process that navigates the Web methodically, starting on a given set of seed pages and following a predefined order. Once the crawler has collected the relevant pages, they are passed on to an information extractor, which obtains the information that is contained in pages and gives it some structure, before returning it back to the

* Supported by the European Commission (FEDER), the Spanish and the Andalusian R&D&I programmes (grants TIN2007-64119, P07-TIC-2602, P08-TIC-4100, TIN2008-04718-E, TIN2010-21744, TIN2010-09809-E, TIN2010-10811-E, and TIN2010-09988-E).

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 228–234, 2012.

user. In this paper, we focus on the crawling aspects of a Virtual Integration process.

As Edwards et al. [10] noted, it is imperative to improve crawlers efficiency in order to adjust to the available bandwidth, specially in Virtual Integration contexts, in which the goal is to retrieve only pages that are relevant to the user query. Consequently, the improvement of crawling efficiency is attracting the interest of many researchers.

The main requirements we consider in the design of a Virtual Integration crawler are efficiency, form filling and unlabelled training sets. The crawler must be efficient, that is, it should minimize bandwidth usage and download only relevant pages. Also, to access pages in the Web that are behind forms, the crawler must be able to fill in forms giving values to the different fields, and submit them. Finally, creating large labeled training sets is burdensome for the user. Instead, we focus on training the crawler using an unlabeled set obtained automatically, minimising user intervention.

Many crawling techniques have been proposed so far in the literature, such as traditional crawlers, focused crawlers or recorders. Traditional crawlers navigate sites by retrieving, analysing and classifying all pages that are found, including non-relevant pages [17]. Hence, they do not fulfill the efficiency requirement. Focused crawlers reduce the amount of irrelevant pages downloaded, usually by applying a content-based web page classifier [1, 7, 6, 9, 12, 14, 15, 16]. They usually deal with large collections of static pages, and therefore form filling is not a priority issue. Finally, recorders are crawlers in which each navigation step has been defined by the user [2, 3, 4, 5, 8, 13, 18]. Therefore, although they deal with form filling and efficiency requirements, the user has to label large training sets.

Our goal is to design a crawler supported by a URL-based classifier to determine page relevance. Thus, it does not need to download a page in order to classify it, reducing the bandwidth and making it efficient and suitable for virtual integration systems. Also, our crawler is able to fill in and submit forms. Furthermore, the crawler includes a setup phase in which a link training set for the classifier is automatically collected, not requiring intervention from the user.

The rest of the article is structured as follows. Section 2 presents the architecture proposed to solve the aforementioned problem; and Section 3 lists some of the conclusions drawn from the research and concludes the article.

2 Architectural Proposal

First, we describe the workflow of the system; then, we present the architectural design, describing for each module a definition of its responsibilities, an example of a typical use case, and a list of the possible issues that should be considered during the design.

2.1 Workflow

Figure 2 presents the system workflow. A Virtual Integration process starts with an enquirer, which translates the user interests into queries that are issued to

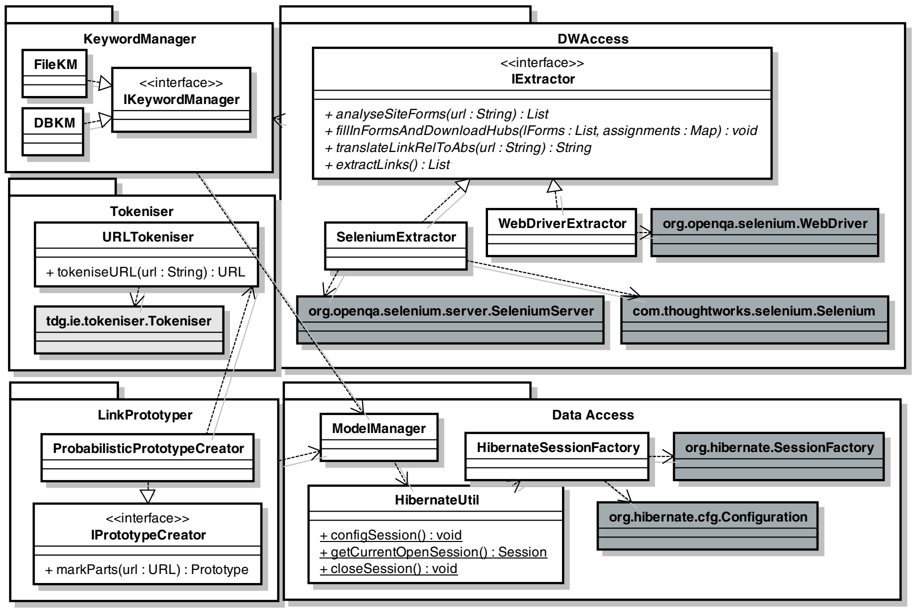


Fig. 1. Class diagram

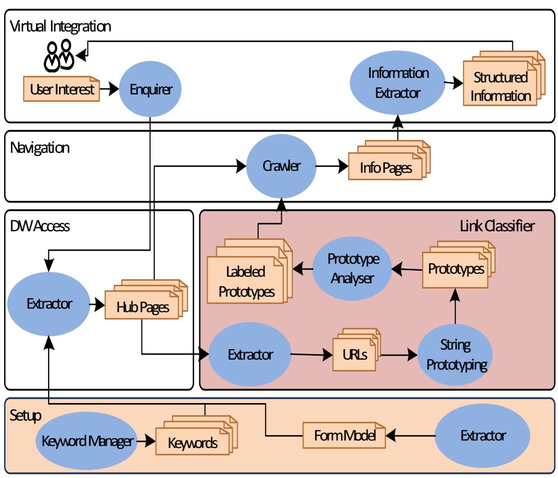


Fig. 2. Workflow diagram

forms. Usually, response pages are hubs, lists of results ordered and indexed, each of them showing a link to another page with detailed information. Relevant pages, when found, are passed on to the information extractor, which obtains and structures the information, that is returned to the user.

We distinguish two phases: the setup and the normal execution phase. In the former, the keyword manager and form filler are focused on obtaining automatically a set of links from the site under analysis, which is later used to train the classifier. The only requirement for the set is to be representative of the site, hence it is extracted from hubs, which are the pages that contain a higher number of links in any given site. In the latter phase, the form filler is used to reach pages behind the forms, and then the crawler makes use of the trained classifier to select which links to follow, avoiding those not leading to relevant pages. In this paper we focus on the setup phase of the architecture, namely in the setup and classifier modules.

2.2 Architectural Design

We present the architecture of our crawler in Figure 1, which includes the main modules in our design: Keyword Manager, Deep Web Access, Tokeniser, String Prototyper and Data Access. Classes of external libraries are highlighted in grey.

The Keyword Manager. The keyword manager is responsible for finding a list of keywords that allow to obtain a representative collection of links when performing the corresponding searches in a given web site. As an example, consider a domain English-spoken web site like Amazon.com. The keyword manager chooses the words that are more likely to appear in the Amazon product database. Given that Amazon store offers a wide product range, a list of the most common English words would suffice to obtain a representative collection of links. Instead, another site like Microsoft Academic Search belongs to a more specific domain, so a list of the most cited authors, for example, would be more useful for this purpose.

The main concerns in this module are related to the language and type of words that are accepted by each site, specially stop words. Stop words tend to have a higher frequency, and they usually yield a higher number of links, but not every site allows searching using stop words. Consider Wordpress.com, a popular blog hosting site, which is unable to find any result related to the keywords 'a' or 'the', while the same words in Youtube.com yield respectively 32,800,000 and 40,000,000 results. Furthermore, stop words may deviate the search and deteriorate results. The lexical type of word is also to be considered, given that verbs are not as frequent as nouns, for example, so they may yield a smaller number of results. Other important factor is the domain to which the site belongs, that defines a specific vocabulary; for example, general, academic, technological or economical, amongst others.

Deep Web Access. The Deep Web Access module is responsible for the interaction with the Deep Web sites, including analysing and filling forms, retrieving response pages, and extracting links from those pages. This module is supported by Selenium Java Library, and it relies on the Keyword Manager that supplies the keywords for form filling. The main interface in this module is IExtractor, which includes all the former methods, and which can be implemented by any suitable library (e.g., Selenium or WebDriver).

For example, to extract information from Amazon.com, the extractor finds the following form in Amazon.com home page:

```
<form action="searchAction" name="site-search">
 <select name="url" id="searchDropdownBox" >
  <option value="aps">...</option>
 </select>
 <input type="text" id="twotabsearchtextbox" name="field-keywords" />
 <input type="image" src="http://images-amazon.com/images/..."/>
</form>
```

The extractor obtains a model that includes the form, with a name attribute with value site-search, and no id attribute, the three fields included in the form, and the submission method, which consists on clicking over the image.

One of the main issues to be solved by the extractor is the lack of standardisation in forms and fields identification. A well defined HTML page should include, at least, an identification attribute for each element, either an id, a name, or both of them, but actually in some web sites we find elements with none of them. In the latter case, the extractor has to deal with the problem of referencing that HTML element for later processing: in the case of a form, to submit it, or to fill it in if the element is a field. We can cite Youtube.com, which is the most visited video sharing web site, as an example of this lack of uniformity. In its home page we can find, amongst others, the following forms:

1. A form with id and no name: <form id="masthead-search" action="/results" onsubmit="...">
2. A form with name and no id <form name="logoutForm" action="/">
3. A form without name or id <form action="/addtoajax">

As for the link extraction, the main issue is that a single URL may be written in different formats, either in relative or absolute form. For example, in Amazon.com, link 1 can be written http://www.amazon.com/ref=logo, /ref=logo or ./ref=logo.

Tokeniser. The Tokeniser module is responsible for processing URLs extracted from the sites pages, and splitting them into tokens, using some configuration. Our implementation of the tokeniser is based on RFC 3986 recommendation for URIs, although not every site conforms to it. Sometimes URLs include special characters, spaces and other symbols that difficult URL parsing. Also, URL query strings are composed of parameters, which may be optional or mandatory, and which may be arranged in different orders.

The String Prototyper. The String Prototyper module is responsible for building a collection of string prototypes, using links extracted from a site, where each prototype is a regular expression.

The main IPrototypeCreator contains method markParts, which discerns variable parts of strings that must be abstracted to create prototypes. An example of implementation for this interface is ProbabilisticPrototypeCreator, which uses a probability-based technique to make the former distinction.

Once the prototypes are built, they are analysed and improved so that their classification results are more accurate. Finally, the prototyper helps the user to assign a label to each prototype, defining the semantic concept contained in the links of the cluster that the prototype represents, and to select relevant concepts. In the Amazon.com example, after processing the prototypes, we obtain the labeled prototypes included in Table 1. More details about the implementation of the prototyper are published at [11].

Table 1. Labeled prototypes obtained from Amazon.com

Label	Prototype (Regular Expression)	Coverage
Reviews	http://www.amazon.com/⋆/product-reviews/⋆?ie=UTF8	30%
Products	http://www.amazon.com/⋆/dp/⋆?ie=UTF8&sr=⋆	30%
Buy New	http://www.amazon.com/gp/offer-listing/ref=olp?ie=UTF8&sr=⋆&condition=new	8%
Buy Used	http://www.amazon.com/gp/offer-listing/ref=olp?ie=UTF8&sr=⋆&condition=used	6%

Data Access. Data access module is responsible for persisting all data in our system. It is based on Hibernate, which manages all objects persistence in a given database (in our case, Oracle). The main classes in this module are HibernateUtil, which contains all methods needed to create and manage a Hibernate session, and ModelManager, which includes methods to manage all objects, including creating, updating and deleting them, and also performing queries to retrieve those objects, using Hibernate support.

3 Conclusions

In this paper, we present an architecture for an efficient crawler that fills in and submits forms, accessing pages in the Deep Web. With respect to the requirements we mentioned in section 1, our proposal classifies web pages depending on the link URL format, so it is applicable to any web site. Hence, our proposal is not only efficient, but also generic and applicable in different domains. Also, our proposal is able to integrate existing form modeling proposals into our crawler, which makes it able to fill in and submit forms, hence dealing with the Deep Web. Finally, our classifier is trained using a set of links collected automatically. The system analyses them and gives the user a list of prototypes representing concepts, while the user is only responsible for defining his or her interest, by labeling and picking one or more prototypes. Not that users intervention is unavoidable, given that the relevancy criteria depends solely on them.

As a result, we designed an efficient crawler, able to access web pages automatically, while requiring as little intervention as possible from the users. A demo of our implementation of the link classifier is available in the author web site [1].

[1] http://www.tdg-seville.info/inmahernandez/CALA+Demo

References

1. Aggarwal, C.C., Al-Garawi, F., Yu, P.S.: On the design of a learning crawler for topical resource discovery. ACM Trans. Inf. Syst. 19(3), 286–309 (2001)
2. Anupam, V., Freire, J., Kumar, B., Lieuwen, D.F.: Automating web navigation with the webvcr. Computer Networks 33(1-6), 503–517 (2000)
3. Baumgartner, R., Ceresna, M., Ledermuller, G.: DeepWeb navigation in web data extraction. In: CIMCA/IAWTIC, pp. 698–703 (2005)
4. Bertoli, C., Crescenzi, V., Merialdo, P.: Crawling programs for wrapper-based applications. In: IRI, pp. 160–165 (2008)
5. Blythe, J., Kapoor, D., Knoblock, C.A., Lerman, K., Minton, S.: Information integration for the masses. J. UCS 14(11), 1811–1837 (2008)
6. Chakrabarti, S., Dom, B., Raghavan, P., Rajagopalan, S., Gibson, D., Kleinberg, J.M.: Automatic resource compilation by analyzing hyperlink structure and associated text. Computer Networks 30(1-7), 65–74 (1998)
7. Chakrabarti, S., van den Berg, M., Dom, B.: Focused crawling: A new approach to topic-specific web resource discovery. Computer Networks 31(11-16), 1623–1640 (1999)
8. Davulcu, H., Freire, J., Kifer, M., Ramakrishnan, I.V.: A layered architecture for querying dynamic web content. In: SIGMOD, pp. 491–502 (1999)
9. de Assis, G.T., Laender, A.H.F., Gonçalves, M.A., da Silva, A.S.: Exploiting Genre in Focused Crawling. In: Ziviani, N., Baeza-Yates, R. (eds.) SPIRE 2007. LNCS, vol. 4726, pp. 62–73. Springer, Heidelberg (2007)
10. Edwards, J., McCurley, K.S., Tomlin, J.A.: An adaptive model for optimizing performance of an incremental web crawler. In: WWW, pp. 106–113 (2001)
11. Hernández, I., Rivero, C.R., Ruiz, D., Corchuelo, R.: A Tool for Link-Based Web Page Classification. In: Lozano, J.A., Gámez, J.A., Moreno, J.A. (eds.) CAEPIA 2011. LNCS, vol. 7023, pp. 443–452. Springer, Heidelberg (2011)
12. Mukherjea, S.: Discovering and analyzing world wide web collections. Knowl. Inf. Syst. 6(2), 230–241 (2004)
13. Pan, A., Raposo, J., Álvarez, M., Hidalgo, J., Viña, Á.: Semi-automatic wrapper generation for commercial web sources. In: Engineering Information Systems in the Internet Context, pp. 265–283 (2002)
14. Pant, G., Srinivasan, P.: Learning to crawl: Comparing classification schemes. ACM Trans. Inf. Syst. 23(4), 430–462 (2005)
15. Pant, G., Srinivasan, P.: Link contexts in classifier-guided topical crawlers. IEEE Trans. Knowl. Data Eng. 18(1), 107–122 (2006)
16. Partalas, I., Paliouras, G., Vlahavas, I.P.: Reinforcement learning with classifier selection for focused crawling. In: ECAI, pp. 759–760 (2008)
17. Raghavan, S., Garcia-Molina, H.: Crawling the hidden web. In: WWW (2001)
18. Wang, Y., Hornung, T.: Deep web navigation by example. In: BIS (Workshops), pp. 131–140 (2008)

A Reference Architecture to Devise Web Information Extractors*

Hassan A. Sleiman and Rafael Corchuelo

Universidad de Sevilla, ETSI Informática,
Avda. Reina Mercedes, s/n, Sevilla E-41012
{hassansleiman,corchu}@us.es
http://tdg-seville.info/

Abstract. The Web is the largest repository of human-friendly information. Unfortunately, web information is embedded in formatting tags and is surrounded by irrelevant information. Researchers are working on information extractors that allow transforming this information into structured data for its later integration into automated processes. Devising a new information extraction technique requires an array of tasks that are specific to this technique and many tasks that are actually common between all techniques. The lack of a reference architectural proposal in the literature to guide software engineers in the design and implementation of information extractors, amounts to little reuse and the focus is usually blurred because of irrelevant details. In this paper, we present a reference architecture to design and implement rule learners for information extractors. We have implemented a software framework to support our architecture, and we have validated it by means of four case studies and a number of experiments that prove that our proposal helps reduce development costs significantly.

Keywords: Information Extraction, Rule Learning Reference Architecture.

1 Introduction

The Web contains a huge amount of information and it is a still growing data container. This unlimited repository aroused enterprises' interests in exploiting this information from the Web by devising new applications that consume and analyse this information. Unfortunately, integrating this information into business processes is a costly task since web information is usually embedded in HTML tags and buried in other superfluous contents. This has motivated many authors to work on web information extractors that allow extracting relevant information from web pages and structuring it in relational tables, which can be easily

* This work was supported by the European Commission (FEDER funds), the Spanish and the Andalusian R&D&I programmes (grants TIN2007-64119, P07-TIC-2602, P08-TIC-4100, TIN2008-04718-E, TIN2010-21744, TIN2010-09809-E, TIN2010-10811-E, and TIN2010-09988-E).

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 235–248, 2012.

consumed by business processes. Information extractors allow extracting information from free-text web pages such as news and blogs, or from semi-structured web pages such as search results and web pages with detailed information about some items. We focus on information extractors from semi-structured web pages.

An information extractor is a general algorithm that can be configured by means of rules to extract the information of interest from a web page and to return it according to a structured model. These rules can be handcrafted [9, 4], predefined as heuristics [3, 5], learnt using supervised techniques that require the user to annotate information of interest in a set of training pages [14, 19], or learnt using non-supervised techniques which learn rules to extract the information the technique considers as data of interest [10, 15]. Common information extraction rules range from regular expressions to context free grammars, first-order rules, XPath templates, and transducers. Our work focuses on techniques that learn transducers to extract information for semi-structured web pages.

The literature reports on few proposals to help researchers and software engineers build their rule learners [2, 8, 13, 20, 17]. These proposals claim to offer an environment in which rule learning techniques can be developed and tested. However, these proposals were presented as tools and the description of their architecture is not clear. Furthermore, they are neither available nor maintained any more. UIMA [12] and Gate [11] are frameworks that can be used to manage large volumes of information, but they both focus on free text pages.

Several proposals referenced the lack of a reference architecture to help software engineers develop extraction rule learners from semi-structured web pages [7, 21, 22]. This is problematic insofar researchers need to implement their proposals from scratch in order to validate them, i.e., they need to pay attention to a variety of details that are ancillary and common to many other proposals, but do not constitute the core of their research [1]. The lack of a reference architecture has also led to a variety of terminologies, which makes communication amongst software engineers difficult, and to experimental results that are not comparable empirically due to differences in the designs and the implementations.

In this paper, we present a reference architecture to devise learners of information extraction rules. To support our reference architecture, we have implemented a software framework and we have validated it by means of four case studies and a series of experiments that prove that it helps reduce development costs significantly and compare the techniques we have implemented empirically. We use the $4 + 1$ architectural view model proposed by Kruchten [18] to describe our reference architecture. In Sections 2–4, we describe the logical, the development, and the scenarios views, respectively. Since our proposal is not intended to be a functioning system, the process and the physical views, which focus on non-functional requirements like concurrency, distribution, topology or communication, do not actually make sense in this case. Section 5 reports on the accompanying framework and on the implementation of several rule learning techniques from the literature, which are compared empirically. Finally, Section 6 concludes our work.

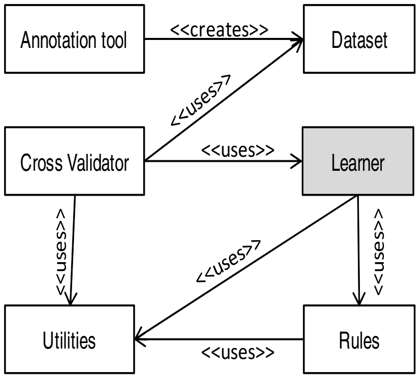


Fig. 1. Relationships amongst subsystems

2 Logical View

The logical view of an architecture represents the functional requirements the system should provide to its end user. In our case, the end user is a software engineer who aims at devising a new extraction rules learning proposal, so in this view we will describe the subsystems, the services they provide, and the interactions amongst them.

The architecture is divided into the following subsystems, whose relationships are shown in Figure 1:

Annotation tool: The reference architecture relies on an annotation tool with which users can download and annotate web pages according to an OWL ontology in which he or she describes classes, properties and their relationships. Ontology classes are used to represent records of information, object properties represent nested records, and data properties represent attributes.

Dataset: This subsystem provides services that allow end users to work with annotations and persist them. During the annotation process, this subsystem allows users to instantiate ontology classes and properties in addition to their position in the corresponding web page. During the learning process, end users can use a dataset to work with a text view or a tree view of the pages they have annotated, get the annotations sorted according to their position or to their type, obtain separating texts between annotations or work with DOM trees and annotation nodes. During the extraction process, this subsystem allows end users to persist the information that is extracted to OWL files.

Learner: This subsystem provides end users with services to develop rule learners. For example, there is a service to create the skeleton of a transducer for a given dataset, i.e., its states and transitions, but not the transition conditions. It saves end users from the burden of inferring the structure of a transducer from

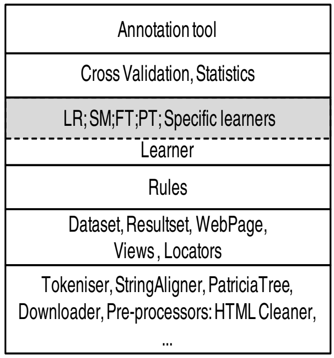


Fig. 2. Reference Architecture layers

the annotations in a dataset, since this is common to every learning algorithm. Note that this subsystem is a point of variability where software engineers only have to focus on devising their own learning algorithms to learn extraction rules.

Rules: This subsystem provides a service to construct extraction rules step by step and to execute them on web pages in order to extract information.

Cross Validator: This subsystem provides a tool with which end users can k-cross validate their rule learners. It helps collect precision, recall, specificity, accuracy, and the $F1$-measure. Thanks to this tool, the results about a given proposal are empirically comparable to other proposals.

Utilities: This subsystem offers some utilities to the rest of subsystems, namely: a configurable tokeniser, a web page downloader, preprocessors such as an HTML cleaner or data region identifier, and a few string and tree alignment algorithms.

3 Development View

This view shows the system from a developer's perspective by illustrating the module organisation of the system and the class diagrams of each module which are the basis for assigning work packages to the members of a development team. Our reference architecture is composed of layers each of which has a well defined responsibility and provides services to the layers above it. These layers are illustrated in Figure 2 and are studied below.

The annotation tool layer: This is the upper layer in our reference architecture. It contains a tool for users to create Datasets. It uses the lower layers to download, clean HTML, add annotations, and to save Datasets. Ontologies are used to create the annotations of each web page.

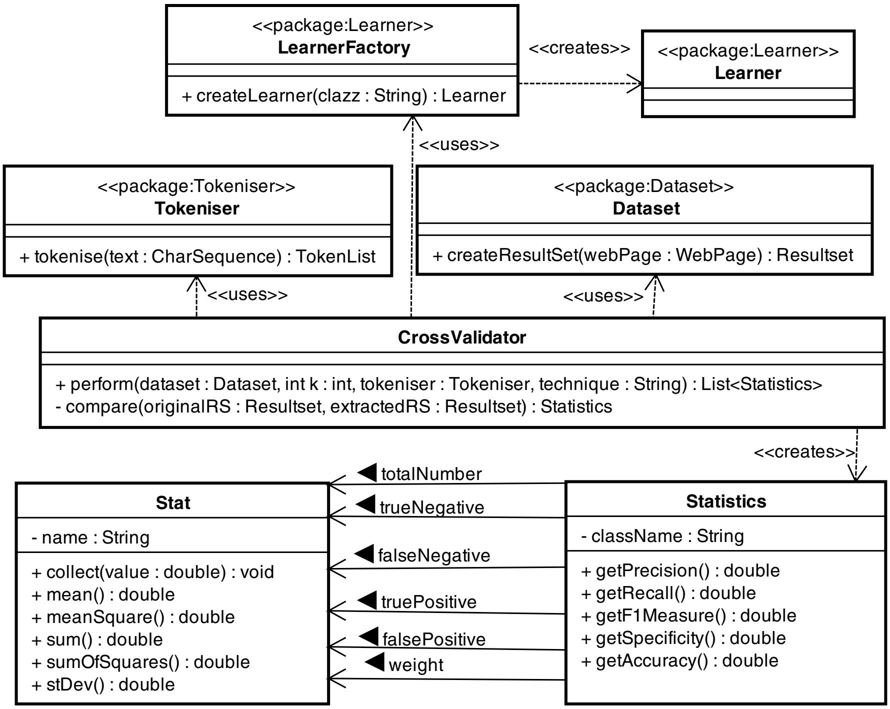


Fig. 3. UML Diagram for the Cross Validation Layer

The cross validator layer: Cross validation is used to estimate the system performance in the practice for a given web site and to obtain comparable results on a Dataset. Given a Dataset obtained from a web site, the k-fold cross validation technique partitions the Dataset into k subsets of web pages with their corresponding Resultsets and then starts iterating over them. At each iteration, it considers one of these subsets for testing, whereas the remaining subsets are considered as a unique set which is used to learn rules. The rules learnt at each iteration are tested on the selected subset, and some statistics are collected. At the end, the weighted arithmetic mean of each calculated value is returned.

The fifth layer contains the the classes of the CrossValidator and the classes to collect results during a k-cross validation Figure 3. These classes are:

– CrossValidator: This class provides methods to perform a k-cross validation on a given a Dataset for a specific Learner. It uses a LearnerFactory to create the Learner to be tested, and during the cross validation process, it compares the Resultsets obtained with the annotated ones to collect the previous Statistics.
– Statistics: This class provides methods to collect the following statistics: true positives, false positives, true negatives, false negatives, the total number of annotations at each iteration and the weight of each class and property in

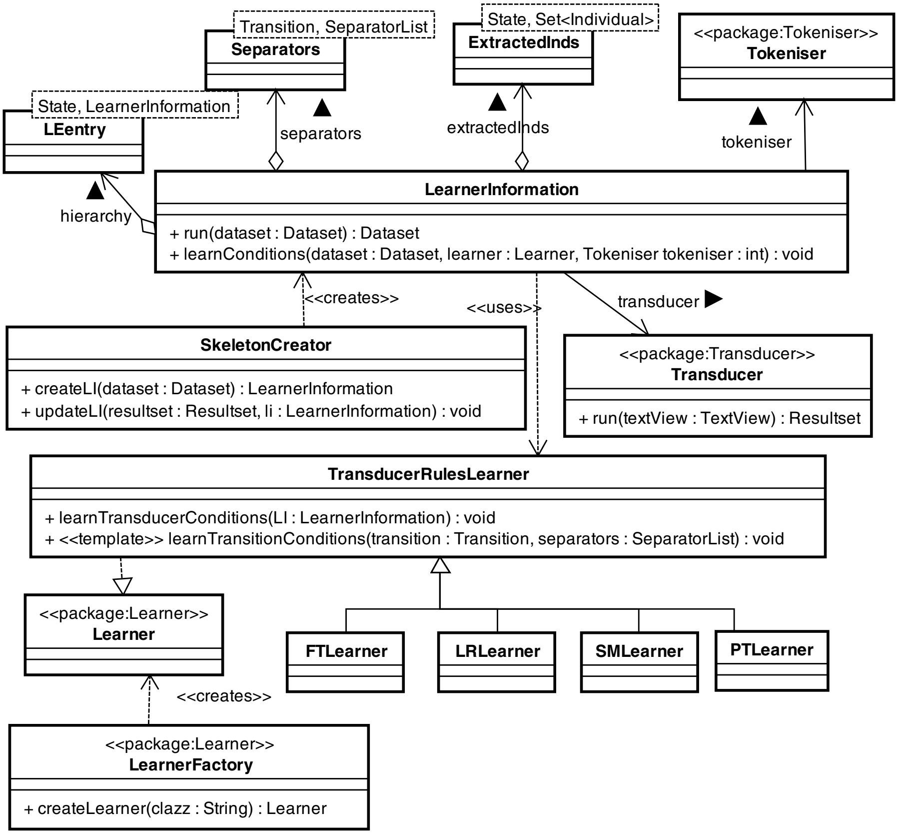


Fig. 4. UML Diagram for the Learners Layer

the Dataset. The provided methods in this class to calculate effectiveness and efficiency values use these attributes.

– Stat: A statistics class in which values can be collected. It provides methods to calculate statistical measures such as arithmetic mean and standard deviation.

The learners layer: The key of this layer is that it is open, i.e., is intended to provide an extension point software engineers should use to create their own rule learners, cf. the gray band in Figure 2. In our accompanying framework we have implemented several extensions to implement learners that learn transducers and that are based on SoftMealy [14], LR [19], FivaTech [15], and IEPAD [6]. The classes in this layer are the following:

– LearnerFactory: This class is intended to provide a method to create a Learner of a given type.

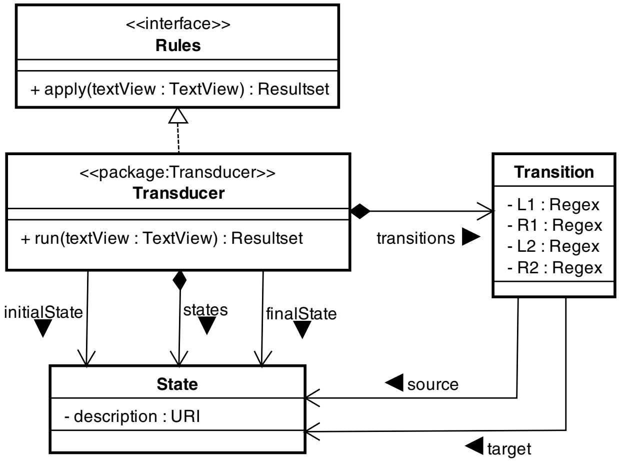


Fig. 5. The UML diagram for the Rules Layer

- Learner: An interface that should be implemented by the rule learners devised by users.
- SkeletonCreator, LearnerInformation, and TransducerRulesLearner : These classes are created to model the rule learners that learn transducers. The SkeletonCreator allows creating the skeleton of a transducer, LearnerInformation models information related to each state and to each transition in the transducer, and the TransducerRulesLearner is intended to learn the transition condition using a specific learning technique.
- FTLearner, LRLearner, PTLearner, and SMLearner: These classes model specific learners that learn transition conditions using techniques inspired in FivaTech [15], LR [19], IEPAD [6], and SoftMealy [14] respectively.

The rules layer: The third layer contains the rules created by learners and used for information extraction. In our accompanying framework we have implemented the rules of type transducers, c.f. 5. The classes in this layer are the following:

- Rules: This is an interface that should be implemented by the different types of extraction rules learnt by the Learners in the upper layer.
- Transducer: This class models a type of rules some information extraction techniques learn. It contains a collection of States and Transitions. Inside each Transition reside its conditions which are regular expressions.

The dataset layer: The second layer contains the dataset classes that help maintain the user annotations and the extracted data. The classes contained in this layer are shown in Figure 6, namely:

- Dataset: It models a collection of Resultsets and web pages. Each contains a map from a set of web pages onto their corresponding Resultsets.

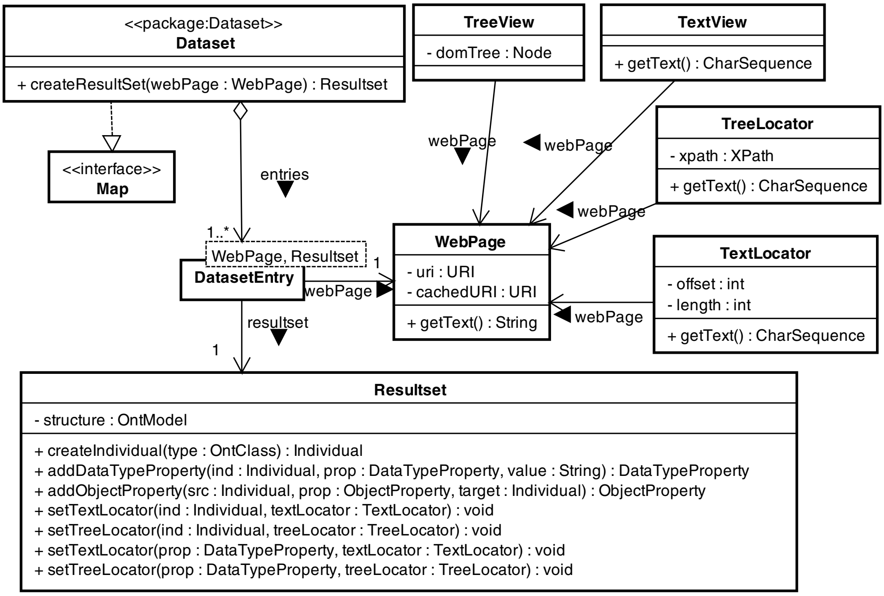


Fig. 6. The Dataset layer

- Resultset: This class models the annotations on a web page and allows to save them as an instance of an ontology.
- WebPage: This class is used to represent a web page. It keeps a reference to where it was downloaded to local cache (cachedURI) and its original location (uri).
- TextLocator and TreeLocator: These classes are used to provide locations to the annotations. Each instance of a class and property in the Resultset has both locators. The TextLocator saves information about the offset and length of an annotation whereas the TreeLocator contains the XPath of the annotation in the DOM Tree. Both locators allow obtaining the text contained in each annotation.
- TextView and TreeView: There classes model views used in the learning process. The TextView provides a view over the text content of a WebPage. The TreeView provides a view over the DOM tree.
- DatasetPersistence: This is a class used to save and load Datasets.

The utility layer: At the bottom of the layers of our architecture resides the utility classes that are used by the upper layers. These utilities include the following classes:

- Tokeniser: A class to implement a configurable tokeniser, cf. Figure 7. The tokeniser is configured by means of an XML file in which hierarchy between different token classes are defined besides the regular expression that defines

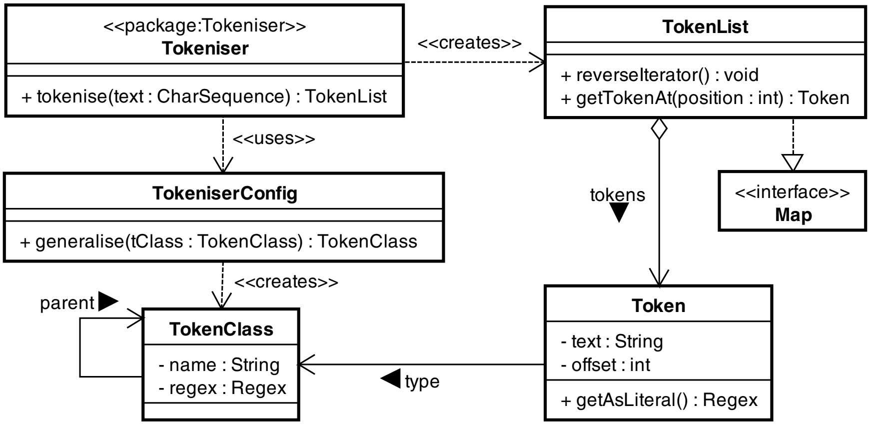


Fig. 7. UML Diagram for the Tokeniser

each type of TokenClass. When a text is tokenised, the Tokeniser returns a TokenList which is a Map of Tokens sorted by their offset in the tokenised text. The methods generalise in TokeniserConfig and getAsLiteral in class Token allow generalisation and specialisation of tokens, respectively.

– StringAligner: This class provides an implementation of a multiple string alignment algorithm that is similar to the one presented in [15].
– PatriciaTree: This class constructs a PatriciaTree starting from a set of token sequences. It provides methods to be updated by adding a new token sequences and to build the regular expression that corresponds to this tree.
– Downloader: This class downloads web pages locally. To ensure the reproducibility of tests, annotated web pages are downloaded and saved locally to avoid that changes in web sites affect user annotations and tests.
– Pre-processors: They are usually used before rule learning or information extraction. HTMLCleaner can be used to fix the HTML code of downloaded web pages while region extractors can be used in some learning techniques to identify the region in which the data of interest resides.

4 Scenarios View

According to Kruchten [18], this view shows how an architecture is instantiated in typical use cases. It serves two purposes: as an illustration of how the architecture can be used and as a validation of the reference architecture since the described scenarios are supposed to be an abstraction of the most important requirements.

SCEN1: Developing a new learning technique: This scenario intends to show how a software engineer can devise his own technique to learn extraction rules of type transducers and integrate it into our reference architecture. The steps he or she should perform are as follows:

- User imports the rule learning framework.
- The user creates a class that extends class TransducerRuleLearner which implements interface Learner.
- A Tokeniser should be created by defining an XML file with the tokenisation hierarchy that will be used by the new technique.
- The user defines the template method learnTransitionConditions in the new class created. This method includes the code necessary to learn the transition conditions for the learnt transducer.

SCEN2: Testing a new learning technique: This scenario intends to show how a software engineer can test his own technique and obtain comparable results by using a 10-folds cross validation, for example. The steps to validate a learning technique is defined below:

- The user downloads our testing Datasets.
- The method perform from the class CrossValidator is called. This method receives $k = 10$, the Tokeniser and the name of the learning technique that will be tested.
- The CrossValidator performs a 10 fold cross validation and save results in Statistics. These Statistics are returned by the CrossValidator and allow to obtain measurements such as Precision and Recall for each class and property in the used ontology.

SCEN3: Learning extraction rules: This scenario describes how user can learn extraction rules for a given web site and save them for future use. The steps to learn these rules, represented as transducers in this case, are the following:

- The user annotates a Dataset using web pages from the web site for which he or she wishes to learn extraction rules. Annotations can be performed using an annotation tool provided with our framework.
- The user should create a SkeletonLearner and call the method create with the Dataset as a parameter. It returns a LearnerInformation object.
- The user creates an object of the new learning technique.
- The user now can learn the transition conditions by calling the method LearnConditions with the Dataset as input.

SCEN4: Applying rules on an input web page: This scenario describes how users can apply extraction rules on web pages to extract information. User should perform the following steps:

- The user loads a Transducer using the TransducerPersistence class.
- The user should create a Resultset for the web page of interest. The extracted information will be saved there.
- The user creates a TextView over the input web page and calls method apply of the transducer with the TextView, a zero to indicate the starting offset, and the Resultset as parameters.

Table 1. Comparing implementation times for NLR, SM, FT and PT

Technique	Using Java 1.6 only	Using our framework	Reduction percentage
NLR	145hrs	32hrs	77.94%
SM	123hrs	87hrs	29.27%
FT	176hrs	61hrs	65.34%
PT	110hrs	30hrs	72.72%

- The transducer runs on this TextView and saves the information extracted into the Resultset. A Dataset with the input WebPage and the final Resultset is created.
- DatasetPersistence should be used now to save the resulting Dataset.

5 Experimental Results

The aim of our reference architecture is to reduce the costs of devising rule learners and to allow comparing learners with each other. To validate it, we have conducted two experiments following the guidelines reported in [16].

First, we have developed a framework to check the viability of our reference architecture. The framework is available for the research community at `http://www.tdg-seville.info/Hassan`. This framework was used to develop a number of the most cited proposals in the literature that are based on transducers or that can be adapted to be used with transducers. We have implemented NLR [19], SM [14], FT [15] and PT [6].

In [16], the authors presented a detailed proposal to guide experimental evaluations. They identified a number of common threats in which we have not incurred:

- Using students in experimental validations: We implemented our techniques with the help of Master Degree students, all of which were junior software engineers who had been working in the industry for a year at least.
- The techniques with which the comparison was performed are not representative: We selected four of the most cited techniques in the literature.
- The experiments are not repeatable: Our proposal and our experiments are available to the research community; anyone can download it, go through the implementation and the documentation and repeat the experiments as long as he or she has a Java 1.6 virtual machine available.
- The results are not comparable: We used 10-fold cross validation, which is the de facto standard to compare experimental results in the field of machine learning.
- Timings are not accurate: We have used the java.lang.management package to measure timings, and every experiment was repeated several times to make sure that the results were accurate. Note that this package allows to measure the time consumed by a single thread, without interferences from other threads or processes.

Table 2. Comparing precision and recall of NLR, SM, FT and PT techniques

Dataset	NLR		SM		FT		PT	
	P	R	P	R	P	R	P	R
soulfilms.com	0.912	0.765	0.993	0.800	1.000	0.000	1.000	0.000
albanianfilmdatabase.com	0.874	0.245	0.874	0.304	0.962	0.371	1.000	0.283
disneymovieslist.com	0.731	0.731	0.989	0.460	1.000	0.000	1.000	0.000
imdb.org	0.753	0.855	0.985	0.845	1.000	0.124	1.000	0.124
citwf.com	0.915	0.915	0.981	0.878	0.992	0.892	0.991	0.343
awesomebooks.com	1.000	0.946	0.830	0.315	1.000	0.676	1.000	0.485
betterworldbooks.com	0.993	0.915	0.877	0.844	0.920	0.514	0.979	0.985
manybooks.net	0.974	0.824	0.746	0.067	0.770	0.536	0.992	0.214

Our first experiment was conducted to check if relying on our reference architecture and an accompanying framework, development costs were reduced remarkably. For this purpose, we requested four postgraduate students with a degree in Software Engineering, to study the previous proposals. Then, they were requested to implement them using Java 1.6. The time to study the requirements, design, develop and test these proposals was measured in hours. Since their development processes were totally independent, each of the participants had to create their datasets for testing.

New postgraduate students were requested to develop the same techniques, but this time using our framework. First, they went through to a training period that was added to the total time that was necessary to study the requirements, design, develop and test the developed techniques. In this case, datasets were reused between the different participants to compare the techniques side by side too. The results of our first experiment are reported in Table 1.

Table 1 shows the time in hours that was necessary to develop and test the techniques on which we report in the first column. The second and the third columns show the time that was necessary to develop these techniques using only Java libraries and using our framework, respectively. The fourth column shows the time reduction for each technique. The costs reduction is clear since the framework allowed reusing components during the development phase and reusing datasets in the testing phase. The last column shows the reduced time percentage; the arithmetic mean of the reduction percentage is 61.31%.

The second experiment was conducted to show the results of the different techniques developed using the framework can be compared now under homogeneous conditions. The implementations obtained in the previous steps were used to perform this empirical study. To the best of our knowledge, this is the first time an empirical comparison between different information extraction techniques, using the same technology and on the same datasets have been reported in literature. Only eight datasets, which were chosen arbitrary, were used in this experiment since our aim is to show how the results are comparable side by side.

Table 2 reports on the results of applying these techniques in practice on several datasets. The first column is the site from which each dataset was obtained. The other columns provide precision (P) and recall (R), which are de facto standard measurements in the information extraction domain. Precision is the percentage of correct information extracted by an information extractor, whereas recall is the percentage of correct information that is successfully extracted. When $P = 1.0$, this means that all the extracted information by the technique is correct, and when $R = 1.0$, this means that all the information that the technique should extract, was extracted correctly. When we apply FT technique over *soulfilms.com* and *disneymovieslist.com* datasets, the technique does not extract any information, which means that all the extracted information is correct and then $P = 1.0$, but none of the correct information is extracted and then $R = 0.0$.

Each dataset contains 30 web pages from the studied site, and the results regarding precision and recall were calculated using 10-fold cross validation. Note that the studied techniques may obtain better results regarding precision and recall by studying sites more thoroughly and adding to the dataset web pages that better represent the web site and that may obtain better extraction rules for this web site, but this is not the focus in this paper.

6 Conclusions

In this paper, we have presented a reference architecture to help software engineers devise new learning techniques in the domain of information extraction. This is the first reference architecture in the literature which provides an abstract, reusable, easy-to-maintain, and easy-to-adapt design that should allow software engineers and researchers face the development of a new rule learning technique without incurring the high costs of developing it from scratch. The reference architecture was validated by an accompanying framework, which was used to show the possibility of developing techniques from literature using our reference architecture, with an overall time reduction that goes beyond 60%.

References

1. Aalst, W., Mylopoulos, J., Sadeh, N.M., Shaw, M.J., Szyperski, C., Lyytinen, K., Loucopoulos, P., Robinson, B., Ralph, P., Wand, Y.: Design Requirements Engineering: A Ten-Year Perspective, vol. 14. Springer, Heidelberg (2009)
2. Adelberg, B.: NoDoSE - a tool for semi-automatically extracting semi-structured data from text documents. In: SIGMOD Conference, pp. 283–294 (1998)
3. Álvarez, M., Pan, A., Raposo, J., Bellas, F., Cacheda, F.: Extracting lists of data records from semi-structured web pages. Data Knowl. Eng. 64(2), 491–509 (2008)
4. Arocena, G.O., Mendelzon, A.O.: WebOQL: Restructuring documents, databases, and webs. TAPOS 5(3), 127–141 (1999)
5. Buttler, D., Liu, L., Pu, C.: A fully automated object extraction system for the world wide web. In: ICDCS, pp. 361–370 (2001)
6. Chang, C.-H., Lui, S.-C.: IEPAD: information extraction based on pattern discovery. In: WWW, pp. 681–688 (2001)

7. Chiticariu, L., Li, Y., Raghavan, S., Reiss, F.: Enterprise information extraction: recent developments and open challenges. In: SIGMOD Conference (2010)
8. Cohen, W.W., Hurst, M., Jensen, L.S.: A flexible learning system for wrapping tables and lists in HTML documents. In: WWW, pp. 232–241 (2002)
9. Crescenzi, V., Mecca, G.: Grammars have exceptions. Inf. Syst. 23(8), 539–565 (1998)
10. Crescenzi, V., Mecca, G., Merialdo, P.: RoadRunner: automatic data extraction from data-intensive web sites. In: SIGMOD Conference, p. 624 (2002)
11. Cunningham, H., Humphreys, K., Gaizauskas, R.J., Wilks, Y.: Software infrastructure for natural language processing. CoRR, cmp-lg/9702005 (1997)
12. Ferrucci, D., Lally, A.: Uima: an architectural approach to unstructured information processing in the corporate research environment. Nat. Lang. Eng. 10, 327–348 (2004)
13. Han, W., Buttler, D., Pu, C.: Wrapping web data into XML. SIGMOD Record 30(3), 33–38 (2001)
14. Hsu, C.-N., Dung, M.-T.: Generating finite-state transducers for semi-structured data extraction from the web. Inf. Syst. 23(8), 521–538 (1998)
15. Kayed, M., Chang, C.-H.: FiVaTech: Page-level web data extraction from template pages. IEEE Trans. Knowl. Data Eng. (2010)
16. Kitchenham, B., Pfleeger, S.L., Pickard, L., Jones, P., Hoaglin, D.C., Rosenberg, J., Emam, K.E.: Preliminary guidelines for empirical research in software engineering. IEEE Trans. Software Eng. 28(8), 721–734 (2002)
17. Kiyavitskaya, N., Zeni, N., Cordy, J.R., Mich, L., Mylopoulos, J.: Cerno: Lightweight tool support for semantic annotation of textual documents. Data Knowl. Eng. (2009)
18. Kruchten, P.: The 4+1 view model of architecture. IEEE Software 12(6), 42–50 (1995)
19. Kushmerick, N.: Wrapper induction: Efficiency and expressiveness. Artif. Intell. 118(1-2), 15–68 (2000)
20. Laender, A.H.F., Ribeiro-Neto, B.A., da Silva, A.S.: DEByE - data extraction by example. Data Knowl. Eng. 40(2), 121–154 (2002)
21. Lavelli, A., Califf, M.E., Ciravegna, F., Freitag, D., Kushmerick, N., Giuliano, C., Romano, L., Ireson, N.: Evaluation of machine learning-based information extraction algorithms: criticisms and recommendations. Language Resources and Evaluation (2008)
22. Madhavan, J., Cohen, S., Dong, X.L., Halevy, A.Y., Jeffery, S.R., Ko, D., Yu, C.: Web-scale data integration: You can afford to pay as you go. In: Conference on Innovative Data Systems Research, pp. 342–350 (2007)

On the Use of Model Transformations for the Automation of the 4SRS Transition Method

Sofia Azevedo[1], Ricardo J. Machado[1], and Rita Suzana Pitangueira Maciel[2]

[1] Universidade do Minho, Portugal
{sofia.azevedo,rmac}@dsi.uminho.pt
[2] Universidade Federal da Bahia, Brazil
ritasuzana@dcc.ufba.br

Abstract. Automation is the essence of MDD (Model-Driven Development). Transforming models into models following a set of rules is at the core of automation. It allows using tools to enliven processes that have been defined. Transition methods are most likely the most important player in the engineering of software. The 4SRS (Four Step Rule Set) is a transition method we adopt in this paper to focus the discussion on the transition from the analysis to the design of software. It has been formalized as a small software development process that can be plugged into larger software development processes. That formalization was conducted with the SPEM (Software & Systems Process Engineering Metamodel), which is a process modeling language for the domain of software and systems. This paper exemplifies how a transition method like the 4SRS can be modeled with the SPEM as a way to study the benefits of the automatic or semiautomatic execution of a transition method as a small dedicated software development process.

Keywords: automation, software process modeling, transition method, software development process, software process modeling language.

1 Introduction

The automation of software processes may be facilitated by process modeling. Method modeling is also essential when it comes to process modeling. The same method can be used by many different processes and more than once in the same process. A method can be defined as a general description of how to develop software and systems. This description shall be the basis for defining or formalizing software and systems development processes. Software and systems development processes can be described as sequences of phases and milestones. The sequence of phases and milestones represents the development lifecycle of the product, so processes may represent product development lifecycles. In this context, methods are only contextualized in a development lifecycle when positioned within a process. Processes can be modeled with process modeling languages like the SPEM (Software & Systems Process Engineering Metamodel) [1]. A process modeling language can be defined as the instrument to express software development processes through a process model [2].

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 249–264, 2012.

The 4SRS (Four Step Rule Set) [3-5] method allows the iterative and incremental model-based transition from user functional requirements (represented as use cases) to system functional requirements (or logical architectures represented as component models). In other words the transformation of use cases (which are dealt with during the analysis of software development) into logical architectures (which are dealt with during the design of software development) is conducted with a method specifically conceived for the purpose. The main concern of the method is: (1) not to lose any user requirements when moving from the analysis to the design; and (2) to assure that no user requirements that have not been elicited with the customer are considered.

A logical software architecture can be faced as a view of a system composed of a set of problem-specific abstractions composed from the system's functional requirements and it is represented as objects or object classes [6]. Another definition of logical software architecture is a module view representing the static structure of the software system (the system's functional blocks, including traceability back to the use cases that express the system's functional requirements) [7]. In the context of this paper, a logical software architecture (represented as a component model) is a design artifact representing a functionality-based structure of the system being designed.

The SPEM (2.0) is a software and systems development process modeling language. We used the SPEM to model the 4SRS as a process. We showed how to model transition methods as processes with a process modeling language (the SPEM) and we used the 4SRS as the example of a transition method to illustrate our approach. In the context of this paper, a transition method is a method that describes how to transform analysis artifacts (use case diagrams) into design artifacts (component models) to develop software. Some other transition methods may for instance describe how to transform design artifacts into implementation artifacts, or how to generate test artifacts, or how to transform business modeling artifacts. Being the 4SRS a method, we formalized it as a small dedicated (at transitioning from analysis to design) software development process that can be plugged into larger software development processes.

The problem this paper addresses is the automation of transition methods, particularly those modeled with the SPEM. The 4SRS was modeled with the SPEM in order to formalize it as a software process. It had to be automated so that it could be enlivened by means of a tool. The SPEM was chosen because it is standard, therefore it would be possible to benefit from the advantages of using a standard that is available to every professional of process modeling. Assuming that disciplines are sets of tasks that can be grouped according to their particular relevance in specific phase(s) from large software development processes into which small dedicated software development processes can be plugged into, transition methods are methods that describe how to transform artifacts from one discipline of a large software development process into artifacts from another discipline of such a process. Transition methods have particularities regarding other methods. They realize a change in the perspective on the system, consequently in the artifacts that represent the system from different perspectives, as well as they mark a change in the phase of the large software development process. In this paper, we use the 4SRS as the example of a transition method modeled with the SPEM and we illustrate the automation of transition methods modeled with the SPEM, in this case a transition method that transforms analysis artifacts into design artifacts. The goal of the automation of transition methods modeled with the SPEM is the automatic or semiautomatic execution of those methods as small dedicated software development processes, in this case the 4SRS. The intent is to decrease the cost of introducing the method into large software development processes, facilitating its use. The (semi)automatic execution of the 4SRS transition method was based on the Moderne [8], which is a tool developed in the Federal University of Bahia (Brazil). The Moderne is a model-driven tool of process modeling and execution. The Moderne tool allows the execution of the 4SRS in an explicit model-driven approach, which implies generating logical architectures through model transformations using a model-to-model transformation language [9, 10].

The paper is structured as follows. Section 2 exposes basics concepts to understand software process modeling, automation, execution and enactment. Section 3 is on work related to process execution through tools, on the SPEM and on process architecture. Section 4 elaborates on a vision over the SPEM and presents the 4SRS transition method. Section 5 shows the preparation necessary for the automation of transition methods modeled with the SPEM, particularly the work undertaken to prepare the automation of the 4SRS. Section 6 presents the case study. Finally Section 7 provides some concluding remarks.

2 Basic Concepts

The goal of processes is to assure the quality of products and the productivity in developing them [11]. Process comprehension and process communication may be negatively affected by the lack of a standard and unified terminology [12]. In such conditions process enactment is far away from process definition, thus the quality of products and the productivity in developing them may be compromised and the goal of processes may not be achieved. Process modeling using a standard and unified terminology suits some purposes like process understanding, process design, process training, process simulation and process support, among others [13]. A short definition of process is the way activities are organized to reach a goal [14]. In the case of software development, a process (software process) can be defined as the set of activities (analysis, design, implementation, testing, among others) organized to deliver a software product (goal). A process model is an artifact that expresses a process with the intention of (among other things) automating that process. A tool supports the execution of the process to consistently reach the goal (delivering a software product).

In 1995 Conradi and Liu [15] says that enactable process models are low-level process models in terms of abstraction. Process modeling languages suit the purpose of detailing process models to make them enactable. According to Henderson-Sellers [16], a process enactment is an instance of a process in a particular project with actual people playing roles, deadlines having real dates and so on. Different enactments have different people playing the same role and different dates for the same deadline.

Bendraou, *et al.* [17] considers that process enactment shall contemplate support for automatic task assignments to roles, automatic routing of artifacts, automatic control on work product states, among others. In what process enactment is concerned, Feiler and Humphrey [18] define an enactable process as an instance of a process definition that shall have process inputs, assigned agents (people or machines that interpret the enactable process), an initial state, a final state and an initiation role. A process definition is a set of enactable process steps. Process definitions can be a composition of subprocess definitions as well as process steps can be a composition of process substeps. A process definition only fits for enactment when fully refined, which means that it cannot be more decomposed into subprocess definitions and into process substeps. The act of creating enactable processes from process definitions is defined by Feiler and Humphrey as process instantiation. They define process enactment as the execution of a process by a process agent following a process definition. Bendraou, *et al.* [17] consider that support for the execution of process models helps coordinating participants, routing artifacts, ensuring process constraints and process deadlines, simulating processes and testing processes. The use of machines in process enactment is called process automation and requires for a process definition to be embodied in a process program [18]. Gruhn [19] defines automatic activities as those executed without human interaction. He mentions that the automation of activities is one of the purposes of process modeling. Another purpose is to govern real processes on the basis of the underlying process models.

3 Related Work

Processes can be executed through tools. In [11] Osterweil considers coding software processes (as part of programming them). Process modeling is one of the parts of programming a software process with a model-driven approach. Software process code is in a lower abstraction level when compared to software process models and it can be executed by computers. Software process code specifications require that software process models define (for new software processes) or formalize (for existing software processes) how software artifacts shall be input to or output from software process tools and how those artifacts are to be handled by the right roles at the right time of the process. Software process models can be analyzed to identify process steps that may be automated. The number of processes being followed to develop software is high. Some key software development processes like software requirements specification and software design lack definition (if they're new ones) or formalization (if they're existing ones). Software design for instance is a process that can be modeled and coded. This paper shows an approach to code a previously modeled software development process, which is the 4SRS transition method modeled with the SPEM as a small dedicated software development process. We analyzed software process models to identify process steps that might be automated with the Moderne tool.

A process' capability can be assessed according to three criteria [20]: task customization, project customization and maturity customization. According to

Henderson-Sellers*, et al.* [20], SPEM allows for task customization and for project customization, but not for maturity customization. SPEM allows for task customization because it allows for the selection of techniques for each task according to the organization and the expertise of the professionals in those techniques. For instance various techniques can be used to elaborate a requirements specification depending on the organization and the project: questionnaires, workshops, storyboards, prototypes and others. SPEM allows for project customization since it allows for the selection of activities, tasks and techniques according to the project. The tasks required to be performed and the products to be developed (models and documents) vary from one project to another. Project customization is a matter of selecting or omitting portions of a process. SPEM does not allow for the addition/removal of activities and tasks to/from a process, and consequently work products depending on the capability or maturity level of the organization. Furthermore Henderson-Sellers*, et al.* [20] refers that SPEM allows for the definition of discrete activities and steps, therefore allowing for process fragment selection.

Kruchten [6] defines development architecture in his “4+1” View Model of Software Architecture. The software system is structured in subsystems that shall be developed by one or a small number of developers (a team). That structure of subsystems is the development architecture, which can be used to allocate work to teams. The development architecture is in fact a logical architecture. In a higher level of abstraction a logical architecture may be process-based (according to our designation) or a process architecture (according to Kruchten’s designation), consisting of a functionality-based structure of the process being designed. To our concern a product-based logical architecture or product architecture is an architecture that resides in a lower level of abstraction comparatively to a process-based or process architecture and consists of a functionality-based structure of the product being designed. Allocating work to teams developing subsystems (as the development architecture can be used to) presupposes that those subsystems can also represent process architecture components that consist of tasks (ultimately steps) that are performed by some roles to produce some output (work products). In fact process architecture components are activities that compose a process structure. Activities are a (kind of) work breakdown element. We can conclude that the 4SRS is not only a method dedicated at transitioning from analysis to design but can also be a method for defining a development architecture (or process architecture). This paper shows how to automate transition methods, particularly those modeled with the SPEM. We use the 4SRS previously modeled with the SPEM as the example of a transition method modeled with the SPEM and we illustrate the automation of transition methods modeled with the SPEM by means of this transition method that transforms analysis artifacts into design artifacts. Automated transition methods modeled with the SPEM can be automatically or semiautomatically executed as small dedicated software development processes. This paper focuses on transition methods from the product development point of view and not from the process architecture point of view.

4 Synopsis of the 4SRS Transition Method

This paper shows how to automate transition methods, particularly those modeled with the SPEM. We use the 4SRS previously modeled with the SPEM as the example of a transition method and we illustrate the automation of transition methods modeled witht the SPEM by means of this transition method that transforms analysis artifacts into design artifacts. Automated transition methods modeled with the SPEM can be automatically or semiautomatically executed as small dedicated software development processes.

The SPEM is a modeling language that contains the minimal elements to define software and systems development processes. The SPEM distinguishes between the concepts of process and of method content. From the process' perspective, activities represent work that can be assigned to roles that intervene on the process, and require inputs and/or outputs (work products) to be performed. Activities are relevant for modeling phases of a development lifecycle (waterfall, iterative and incremental are three types of development lifecycles). The same role can be used in an early phase of a development lifecycle represented by an activity and in a later phase of the same development lifecycle represented by another activity, which may mean that e.g. that role will handle different work products in those two different phases.

A method content can be considered as a set of concepts (tasks, roles and work products) that allows representing a software and systems development method and its techniques which can be positioned within a specific software and systems development lifecycle (consider that a method is composed of many techniques and that a development lifecycle can be composed of a sequence of phases and milestones for example). Processes shall use method content elements and organize them into sequences. A method content is a stepwise definition of tasks that shall be performed by roles to originate work products. They may consume work products as well. A task from a method content may have its steps, inputs or outputs (work products) changed depending on the development lifecycle it is positioned.

The main difference between a method content and a process is that a method content defines methods and techniques for software and systems development processes, whereas a process defines the positioning of those methods and techniques within a development lifecycle composed of e.g. a sequence of phases and milestones. When a specific composition of tasks, roles and work products (a specific method content) is positioned within a development lifecycle, it means that the method content has been applied to that part of the process where it has been positioned. It also means that the method content was used by that process.

The 4SRS is a method for obtaining system functional requirements from user functional requirements [3]. Use cases model user functional requirements and logical architectures model system functional requirements. Use cases are problem-related, technology-independent and are dealt with during the analysis phase of software development. Logical architectures are solution-related, technology-independent and are dealt with in the beginning of the design phase of software development.

According to Kaindl [21], we can classify 4SRS as a transition method. Kaindl argues that it is difficult to move from the analysis to the design of software. From the perspective of object-oriented software development, the main reason is that analysis objects and design objects represent different kinds of concepts. Analysis objects are from the problem domain and represent objects from the real world. Design objects are from a solution domain and shall indicate how the system shall be developed. Design objects are abstractions of code or the implementation details needed in order to build a system with that solution to that problem. Design objects are both an abstraction of concepts from the problem domain and of the implementation of the system to be built. An analysis model can become part of a design model by influencing architectural decisions. The 4SRS is a method that allows moving from the analysis to the design of software. In the case of the 4SRS, the analysis model (a UML (Unified Modeling Language) [22] use case diagram) influences architectural decisions that originate the design model (a UML component model).

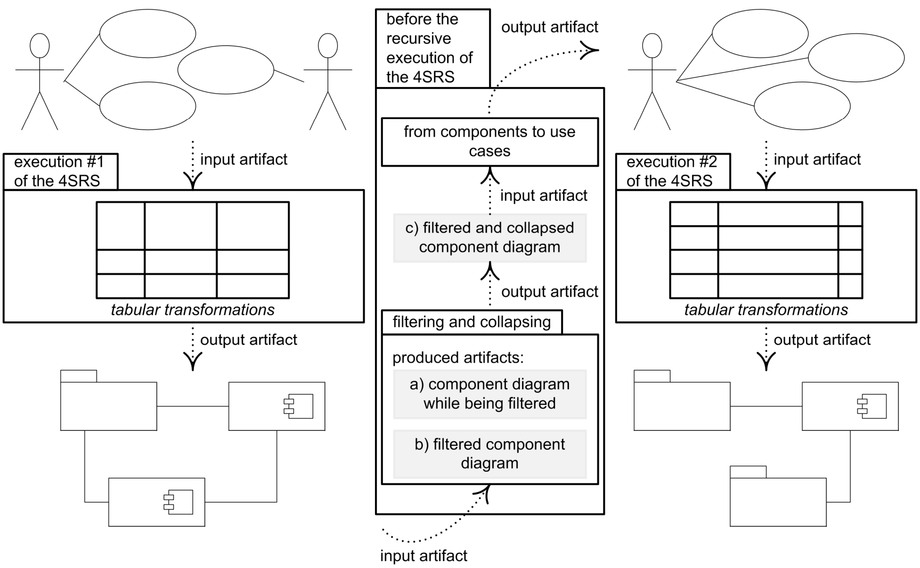


Fig. 1. Schematic representation of the recursive execution of the 4SRS method

Shortly the 4SRS method is composed of the following steps: (1) *Component Creation*, to create three kinds of components for each use case (an interface component, a control component and a data component; other kinds of components could be created, so this is not a limitation of the method, rather an architectural decision); (2) *Component Elimination*, to remove redundant requirements and find missing requirements (this step is vital in order to validate the components blindly created in the previous step and includes eliminating the components whose requirements are already represented by other components; the finding of missing requirements means that components have been inadequately eliminated or use cases are missing); (3) *Component Packaging and Nesting*, to semantically group components in packages; and (4) *Component Association*, to define associations of components with each other in the component model.

Architectural refinement is the approach the 4SRS method takes to increment a primary logical architecture with detail (by primary we mean the architecture that is going to be detailed) [4]. In the context of this paper, recursion is the ability of the method to be executed over parts of its output artifact after transformed into the input artifact for that execution. As depicted in Figure 1, the 4SRS method may be applied recursively, in several *executions*. In the context of each one of those executions, various *iterations* can be performed. Although there is no stopping rule for iterating over the same set of use case diagrams, it shall be performed until the results obtained generate a logical architecture that does not benefit from additional iterations in terms of the elimination of redundant requirements, the finding of missing requirements and the increasing of the logical architecture's cohesion. There cannot be components isolated from the rest of the architecture when the global architecture is composed from the various logical architectures generated by the different executions. In the case of refinement (by recursion), when one of the executions is considered to be finished by the modeler, the output of that execution's last iteration (a component model) is going to originate the input of a subsequent execution's first iteration (a use case diagram). The task flow of the new execution is exactly the same as the task flow of the preceding one. Again in the case of refinement (by recursion), the logical architectures produced by the various executions are situated in lower levels of abstraction and cover less functionality than the logical architectures they refined.

Considering architectural refinement, the sequence of steps for the 4SRS method is the following: (1) *Component Creation*; (2) *Component Elimination*; (3) *Component Packaging and Nesting*; (4) *Component Association*; (4+1) *Filtering and Collapsing*; and (4+2) *From Components to Use Cases*. The first four steps are the original steps and the other ones are what we call the *intermediate steps*, which are performed in between executions of the 4SRS method. The step 2 is composed of seven microsteps. The microstep (2.i) *Use Case Classification* is about determining the kinds of components that will originate from each use case according to the eight possible combinations. According to this classification, the microstep (2.ii) *Local Elimination* is about eliminating the components blindly created in the step 1 by analyzing the textual description of the use cases and deciding on whether those components make sense in the problem domain. The microstep (2.iii) *Component Naming* is about naming the components that have not been eliminated in the previous microstep. The microstep (2.iv) *Component Description* is about textually describing the components named in the previous microstep, based on the textual descriptions of the use cases they originated from, on nonfunctional requirements and on design decisions. The microstep (2.v) *Component Representation* is about determining whether some components represent both their own system requirements and others'. The microstep (2.vi) *Global Elimination* is about eliminating the components whose requirements are already represented by other components (elimination of functional redundancy). Finally the microstep (2.vii) *Component Renaming* is about renaming the components that were not eliminated in the previous microstep and that represent additional components.

The filtering consists of considering some components as the subsystem for refinement and discarding those that are not associated with them [5]. The collapsing consists of hiding the details of the subsystem whose components are going to be refined. Later on, those components are replaced inside the limits of the subsystem's border by others of lower abstraction level.

The intermediate step (4+2) *From Components to Use Cases* of the 4SRS method is composed of two intermediate substeps: the (4+2.i) *Deriving Use Cases from Components* and the (4+2.ii) *Detailing Use Cases*. The goal of the (4+2.i) *Deriving Use Cases from Components* is to derive the use cases to hand out as input for the succeeding recursive execution of the 4SRS method from the components to refine. The goal of the (4+2.ii) *Detailing Use Cases* is to refine those use cases.

5 Automating the 4SRS Method

The goal of automated transition methods modeled with the SPEM is to automatically or semiautomatically execute them as small dedicated software development processes. That goal was achieved with the support of a tool. That tool is the Moderne, which is a tool developed at the Federal University of Bahia (Brazil). The Moderne is a model-driven tool of process modeling and execution. The Moderne tool allows the execution of the 4SRS in an explicit model-driven approach, which implies generating logical architectures through model transformations using a model-to-model transformation language. These model transformations can be executed with any ATL (Atlas Transformation Language) [9] engine that uses UML, and not only with the Moderne.

This section exposes the way the 4SRS transition method modeled with the SPEM has been automated according to our definition of goal for the automation of transition methods modeled with the SPEM: the automation allows the automatic or semiautomatic execution of these transition methods as small dedicated software development processes. By automatic it is meant that models (the artifacts) are transformed using a transformation language or based on some action the modeler (the tool user) performs with the tool (to which the tool is programmed to respond) or even based on rules the tool has been programmed with to respond to some particular event without any modeler's action. By semiautomatic it is meant that the tool supports decisions the modeler has to make by allowing him to represent them in the diagrams.

The modeling of the 4SRS transition method with the SPEM we performed beforehand (and that is not the focus of this paper) had to be adapted in order for the method to be automatically or semiautomatically executed as a microprocess with the Moderne tool. In the context of this paper, a microprocess is a small dedicated software development process dedicated at transitioning from analysis to design and that can be plugged into larger software development processes. From the perspective of the SPEM, a process can be considered to be at least a method content (or shortly method) positioned within a development lifecycle. In this paper, a method defines tasks (composed of steps), roles and work products, therefore methods are modeled with the following elements: tasks (and steps), roles and work products. We subclassed tasks into transition tasks and intermediate tasks, steps into transition steps and intermediate steps, and finally work products into initial work products, intermediate work products and final work products. Figure 2 illustrates some examples of these elements. In the case of tasks, steps and work products, the stereotypes respectively indicate the type of task, step or work product according to the subclassing just mentioned.

The model of the 4SRS transition method with the SPEM (elaborated beforehand) was adapted by adding the transformation rule *i-c-dComponentsFromLeafUseCases* as an input to the transition task *ComponentCreation* and by adding the intermediate task *UseCaseDiagramDrawing* to the model. The transformation rule is an ATL [9] rule that defines how to transform the use case diagram (the initial work product *UseCaseDiagram*) into the component diagram (the intermediate work product *i-c-dComponents*). The intermediate task had to be modeled to give the input (initial work product *UseCaseDiagram*) to the task that consumes the only initial work product in the model of the 4SRS transition method with the SPEM, which are the transition task *ComponentCreation* and the initial work product *UseCaseDiagram*. The intermediate task *UseCaseDiagramDrawing* was needed since the Moderne tool does not allow creating an input to a task in the context of the task itself, rather in the context of another task as output of that own task.

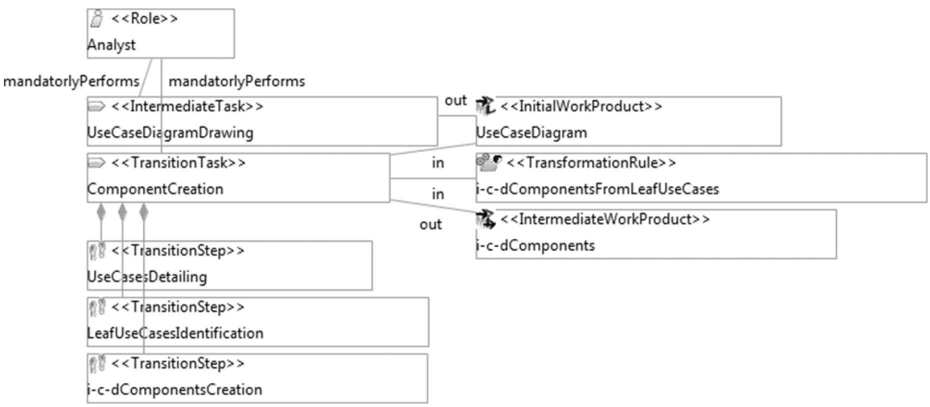


Fig. 2. The 4SRS transition method modeled with the SPEM for automation purposes

We analyzed the model of the 4SRS transition method with the SPEM to identify the steps that could be automated with the Moderne tool. Table 1 shows that analysis. Some of the steps from the 4SRS transition method where concluded to be fully automated with the Moderne tool whereas others where concluded to be semiautomated or not automated at all. The automation capability is the ability of a method's step to be automated with a tool. The ones concluded to be fully automated with the tool were classified as "Automatic" in terms of their automation capability, the ones concluded to be semiautomated with the tool were classified as "Semiautomatic" and the ones concluded to be not automated with the tool where classified as "Not automatic". The automatic steps were automated using ATL model-to-model transformation rules. A semiautomatic step depends on some modeler's action in the models by means of the tool before the ATL model-to-model transformation rules concerning that particular step can be applied. The not automatic steps comprise actions that are fully performed by the modeler even that they consist of input for the models or for the information attached to the models, in the tool.

6 Case Study

The use case diagram in Figure 3 was used to exemplify the model-to-model transformations the Moderne tool is able to perform in the context of the 4SRS. The diagram is based on the Fraunhofer IESE's GoPhone [23] case study, which presents a series of use cases for a part of a mobile phone product line particularly concerning the interaction between the user and the mobile phone software.

Table 1. Analysis of the automation capability of the steps from the 4SRS

4SRS Step/Microstep	Automation Capability
Step 1: Component Creation	Automatic
Microstep 2.i: Use Case Classification	Not automatic (the modeler shall decide each use case's classification)
Microstep 2.ii: Local Elimination	Semiautomatic (the modeler shall tag in the component diagram the components to eliminate or maintain)
Microstep 2.iii: Component Naming	Not automatic
Microstep 2.iv: Component Description	Not automatic
Microstep 2.v: Component Representation	Not automatic (the modeler explicitly relates components in the diagram through Dependency relationships indicating which component represents others)
Microstep 2.vi: Global Elimination	Automatic (based on the Dependency relationships mentioned above in this table)
Microstep 2.vii: Component Renaming	Not automatic
Step 3: Component Packaging and Nesting	Not automatic
Step 4: Component Association	Semiautomatic (partially based on the rules for associating components and partially based on the modeler's decision)
Intermediate step 4+1: Filtering and Collapsing	Semiautomatic (the collapsing is automatic; the filtering is semiautomatic depending partially on the modeler's decision to perform refinement and with the automatic exclusion of the components not associated with any component from the region to refine determined by the modeler)
Intermediate microstep 4+2.i: Deriving Use Cases from Components	Not automatic
Intermediate microstep 4+2.ii: Detailing Use Cases	Not automatic

The transformation of the use case diagram in Figure 3 into the corresponding component diagram was defined in an ATL rule that mostly determines what leaf use cases are. Leaf use cases are those from which interface components, control components and data components are generated in the step 1 of the 4SRS (Component Creation). The ATL rule defines that leaf use cases are those that are included by at least one use case and that do not include any other use case, and those that are not included by any use case and do not include any use case. The rule also defines some associations between components, and between components and actors (from the use case diagram). This anticipates part of the step 4 (Component Association) of the 4SRS to the step 1 (Component Creation).

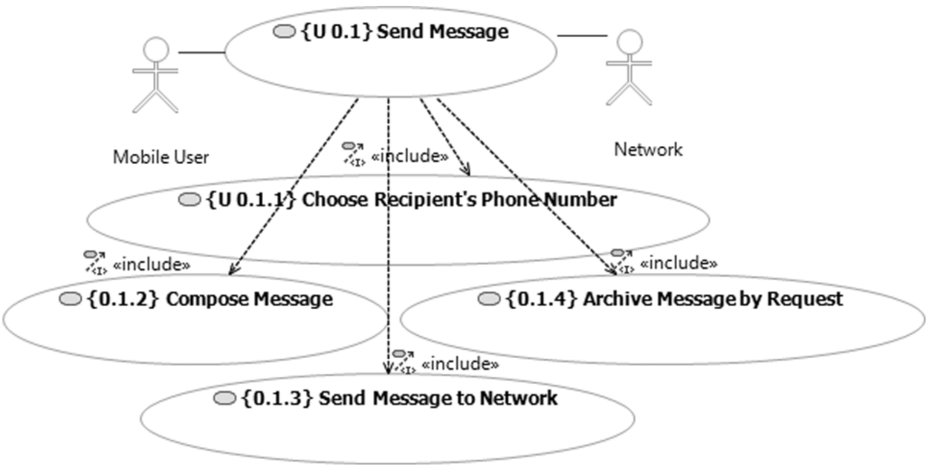


Fig. 3. A use case diagram from the GoPhone

Figure 4 depicts part of the ATL rule that determines the leaf use cases of a use case diagram. The function `srcIncludes()` gets all associations whose source is the element that called the rule. The component diagram generated from the use case diagram in Figure 3 through the ATL rule and the Moderne tool is in Figure 5.

```
helper context UML!UseCase def : isLeaf() : Boolean =
       self.srcIncludes()->size() = 0;
```

Fig. 4. Part of the ATL rule that determines the leaf use cases of a use case diagram

We have defined some well-formedeness rules or constraints in OCL (Object Constraint Language [24]) to do what the following figures illustrate.

A transition task can transform an initial work product into an intermediate work product or an intermediate work product into an intermediate work product or even an intermediate work product into a final work product. The OCL code for these constraints is in Figure 6. Figure 7 shows a validation error signaled with a cross in a transition task that transforms an intermediate work product into an initial work product.

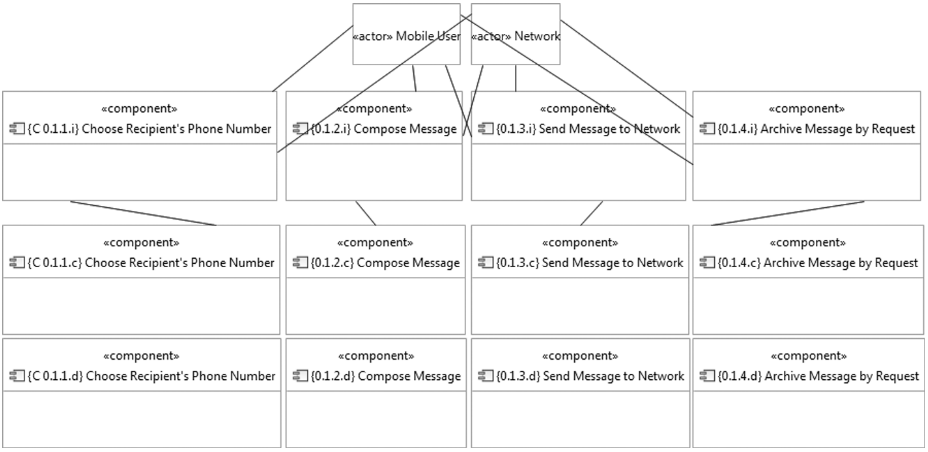


Fig. 5. The component diagram automatically generated from the use case diagram in Figure 3

```
context TransitionTask
inv:
(self.in->forAll(wp | wp.oclIsTypeOf(InitialWorkProduct))
and
self.out->forAll(wp | wp.oclIsTypeOf(IntermediateWorkProduct)))
OR
(self.in->forAll(wp | wp.oclIsTypeOf(IntermediateWorkProduct))
and
self.out->forAll(wp | wp.oclIsTypeOf(IntermediateWorkProduct)))
OR
(self.in->forAll(wp | wp.oclIsTypeOf(IntermediateWorkProduct))
and
self.out->forAll(wp | wp.oclIsTypeOf(FinalWorkProduct)))
```

Fig. 6. The OCL code for constraints on the relation between transition tasks and work products

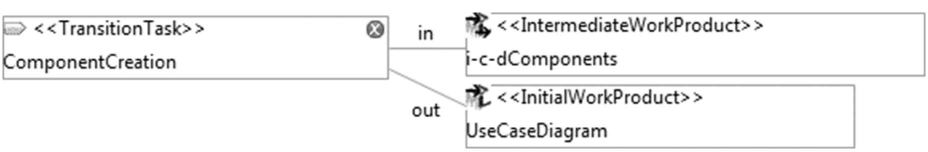


Fig. 7. An example of a validation error in the constraints on the relation between transition tasks and work products

An intermediate task transforms a final work product into an initial work product. The OCL code for this constraint is in Figure 8. Figure 9 illustrates a validation error signaled with a cross in an association between an intermediate task and an initial work product.

```
context IntermediateTask
inv:  self.in->forAll(wp | wp.oclIsTypeOf(FinalWorkProduct));
inv:  self.out->forAll(wp | wp.oclIsTypeOf(InitialWorkProduct));
```

Fig. 8. The OCL code for constraint on the relation between intermediate tasks and work products

Fig. 9. An example of a validation error in the constraint on the relation between intermediate tasks and work products

A transition step can only be contained by a transition task, whereas an intermediate step can only be contained by an intermediate task. The OCL code for these constraints is in Figure 10. Figure 11 depicts that a composition could not be drawn between a transition task and an intermediate step. In the case of the constraints above, a validation error was signaled with a cross in model elements because changing properties (the names) of the associations would eliminate that error. In this case changing properties of the association would not eliminate the error since the association should not exist in the first place to obey the constraint.

```
context IntermediateTask
inv: steps->forAll(step | step.oclIsTypeOf(IntermediateStep))

context TransitionTask
inv: steps->forAll(step | step.oclIsTypeOf(TransitionStep)))
```

Fig. 10. The OCL code for constraints on the relation between tasks and steps

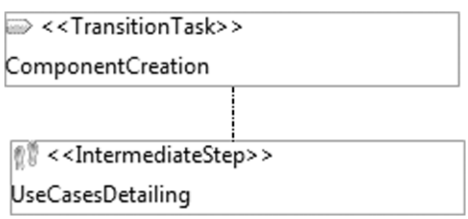


Fig. 11. An example of the impossibility of a composition between a transition task and an intermediate step

7 Conclusions

Transition methods describe how to transform artifacts originally produced within a certain discipline of a large software development process into artifacts from another discipline of such a process. Some transition methods are targeted at moving from the analysis to the design of software. The 4SRS is a method that allows transforming UML use case diagrams (considered here as the analysis model) into the logical architecture of the system as a UML component model (considered here as the design model, the first technical artifact to initiate the design of the system).

This paper describes the automation of a transition method, using the 4SRS method modeled with the SPEM as the case study. We adopted the Moderne tool to automate the generation of logical architectures through model transformations defined with the ATL language. We have defined some well-formedeness rules or constraints in OCL to validate the modeling of the 4SRS transition method with the SPEM.

The SPEM model of the 4SRS transition method elaborated beforehand was adapted for automation purposes. For instance a transformation rule was added to the model.

The OCL constrains on the modeling of the 4SRS transition method with the SPEM established the well-formedeness of the relations between transition tasks and work products, between intermediate tasks and work products, and between tasks and steps (all of these are elements for modeling methods). The violation of the constraints on those relations was tagged in the model through validation errors.

In terms of future work, we plan to assess the efficiency of our approach by adopting the Moderne tool to apply the 4SRS transition method in a real industrial project.

References

[1] OMG. Software & Systems Process Engineering Meta-Model Specification - version 2.0 (2008), `http://www.omg.org`

[2] Conradi, R., Jaccheri, M.L.: Process Modelling Languages. In: Derniame, J.-C., Kaba, B.A., Wastell, D., et al. (eds.) Promoter-2 1998. LNCS, vol. 1500, pp. 27–52. Springer, Heidelberg (1999)

[3] Machado, R.J., et al.: Transformation of UML Models for Service-Oriented Software Architectures. Presented at the 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, ECBS 2005, Greenbelt, Maryland, USA (2005)

[4] Machado, R.J., Fernandes, J.M., Monteiro, P., Daskalakis, C.: Refinement of Software Architectures by Recursive Model Transformations. In: Münch, J., Vierimaa, M., et al. (eds.) PROFES 2006. LNCS, vol. 4034, pp. 422–428. Springer, Heidelberg (2006)

[5] Fernandes, J.M., et al.: A Demonstration Case on the Transformation of Software Architectures for Service Specification. Presented at the 5th IFIP Working Conference on Distributed and Parallel Embedded Systems, DIPES 2006, Braga, Portugal (2006)

[6] Kruchten, P.: Architectural Blueprints - The "4+1" View Model of Software Architecture. IEEE Software 12, 42–50 (1995)

[7] Clements, P., et al.: Documenting Software Architectures: Views and Beyond. Addison-Wesley, Upper Saddle River (2002)

[8] Gomes, R.A., et al.: Moderne: Model Driven Process Centered Software Engineering Environment. Presented at the 2nd Congresso Brasileiro de Software: Teoria e Prática, CBSoft 2011, São Paulo, Brasil (2011)

[9] The Eclipse Foundation. ATL Project (2010), http://www.eclipse.org/m2m/atl

[10] Brown, A.W., et al.: Introduction: Models, Modeling, and Model-Driven Architecture (MDA). In: Beydeda, S., et al. (eds.) Model-Driven Software Development, pp. 1–16. Springer, Heidelberg (2005)

[11] Osterweil, L.J.: Software Processes Are Software Too, Revisited: An Invited Talk on the Most Influential Paper of ICSE 9. Presented at the 1997 International Conference on Software Engineering, ICSE 1997, Boston, Massachusetts, USA (1997)

[12] Maciel, R.S.P., et al.: An Integrated Approach for Model Driven Process Modeling and Enactment. Presented at the XXIII Brazilian Symposium on Software Engineering, SBES 2009, Fortaleza, Brazil (2009)

[13] Fuggetta, A.: Software Process: A Roadmap. Presented at the 22nd International Conference on on Software Engineering, ICSE 2000, Limerick, Ireland (2000)

[14] Estublier, J.: Software are Processes Too. In: Li, M., Boehm, B., Osterweil, L.J. (eds.) SPW 2005. LNCS, vol. 3840, pp. 25–34. Springer, Heidelberg (2006)

[15] Conradi, R., Liu, C.: Process Modelling Languages: One or Many? In: Schäfer, W. (ed.) EWSPT 1995. LNCS, vol. 913, pp. 98–118. Springer, Heidelberg (1995)

[16] Henderson-Sellers, B.: Process Metamodelling and Process Construction: Examples Using the OPEN Process Framework (OPF). Annals of Software Engineering 14, 341–362 (2002)

[17] Bendraou, R., et al.: A Comparison of Six UML-Based Languages for Software Process Modeling. IEEE Transactions on Software Engineering 36, 662–675 (2010)

[18] Feiler, P.H., Humphrey, W.S.: Software Process Development and Enactment: Concepts and Definitions. Presented at the 2nd International Conference on the Software Process, ICSP 1993, Berlin, Germany (1993)

[19] Gruhn, V.: Process-Centered Software Engineering Environments, A Brief History and Future Challenges. Annals of Software Engineering 14, 363–382 (2002)

[20] Henderson-Sellers, B., et al.: Process Construction and Customization. Journal of Universal Computer Science 17, 326–358 (2004)

[21] Kaindl, H.: Difficulties in the Transition from OO Analysis to Design. IEEE Software 16, 94–102 (1999)

[22] OMG. Unified Modeling Language: Superstructure - version 2.2 (2009), http://www.omg.org

[23] Muthig, D., et al.: GoPhone - A Software Product Line in the Mobile Phone Domain. Fraunhofer IESE, IESE-Report No. 025.04/E (March 5, 2004)

[24] OMG. Object Constraint Language: Specification - version 2.2 (2010), http://www.omg.org

Model Driven Workflow Development with $T_\Box$

Fazle Rabbi and Wendy MacCaull

Centre for Logic and Information,
St. Francis Xavier University
{rfazle,wmaccaul}@stfx.ca

Abstract. Model Driven Engineering (MDE) refers to the systematic use of models as primary engineering artifacts throughout the engineering lifecycle. MDE has a lot of potential to make adaptive software systems, but it requires maturity and tool support. Here we present a domain specific language, called $T_\Box$ (pronounced as T-Square) for writing workflow process specifications which allows us to write procedural statements for tasks and branch conditions, to query an ontology and to declare user interfaces. We apply transformation methods to generate executable software from the abstract process specifications.

Keywords: Workflow Management System, Model Driven Engineering, Ontology, Domain Specific Language, Adaptive System.

1 Introduction

Software researchers and developers require abstractions of their system to help them program in terms of their design intent rather than in terms of the underlying computing environments e.g., CPU, memory, and network devices [18]. Although early programming languages, such as assembly and Fortran, raised the level of abstraction by shielding developers from complexities of programming with machine code, they still had distinct "computing oriented" focuses. Advances in languages and platforms during the past two decades have minimized the need to reinvent common and middleware services, such as transactions, discovery, fault tolerance, event notification, security, and distributed resource management, by providing libraries, APIs, etc. But programmers and developers still need to focus on the use of those libraries, APIs, services, etc. Model Driven Engineering (MDE) further raises the level of abstraction in program specification and aims to increase automation in software development [18]. MDE offers a promising approach to alleviate the complexity of platforms by expressing domain concepts effectively by models. A model is specified by modeling notations or modeling languages. Since modeling languages are usually tailored to a certain domain, such a language is often called a Domain Specific Language (DSL). A DSL can be visual (e.g., UML, BPMN [1]) or textual (e.g., CSS, regular expressions, ant, SQL). DSL helps developers focus on a problem domain rather than on technical details [14].

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 265–279, 2012.

Today business process models are frequently used to describe the behaviour of large systems with characteristics like concurrency, resource sharing, and synchronisation [11]. Many of today's workflows are complex requiring a high degree of flexibility, and massive data and knowledge management. In this paper we present a DSL called $T_\square$ (pronounced T-Square) for writing workflow process specifications, which incorporates the following features: a) a simple syntax for i) writing procedural statements, ii) querying and manipulating ontologies, iii) designing rich user interfaces (UIs); b) abstraction of communication details; and c) ease of customization.

Ontologies are used in $T_\square$ for data and knowledge persistence. Software modelling languages and methodologies can benefit from the integration with ontology languages in various ways, e.g., by reducing language ambiguity, enabling validation and automated consistency checking [21]. Moreover intelligent applications may be built based on ontology reasoning. In addition to providing a high level syntax with an automatic translation to platform specific code, this DSL benefits from the use of ontologies [20] which allows the user to infer knowledge, further reducing the coding effort, making $T_\square$ suitable for rapid application development.

This work is part of research project, "Building decision–support through dynamic workflow systems for healthcare", a collaboration among academic researchers, a local health authority and an industry partner [12]. The goal is to develop tools for process, communication and information for community based healthcare programs. Workflow tools to guide process need to interface with a complex healthcare knowledge base and be supported by electronic forms for capturing patient information, making it accessible and usable by a variety of health providers, in a variety of service settings. While healthcare is generally governed by national or provincial standards, a change in regulations or local conditions - e.g., at the clinic, hospital or doctor's office - requires that the workflow and/or task specification easily adapt to comply with the change. Forms, in particular, are often local to a setting and may need to be modified quickly and/or frequently, to meet the needs of clinicians or administrators wishing to track some disease or specific aspect of care. Data must be stored, aggregated, interpreted and reasoned with, both locally and nationally. Ontologies are becoming the accepted method for standardizing and storing machine computable information. Healthcare is a safety critical process: technology to generate reliable software is essential.

In section 2 we propose the model driven approach for workflow development; in section 3 we give details of the proposed language, $T_\square$; in section 4 we include some related work, and in section 5 we conclude the paper with some directions for future work.

2 Model Driven Workflow Development

Workflow systems are often designed with graphical workflow modeling languages such as Business Process Modeling Notation (BPMN) [1], Yet Another

Workflow Language (YAWL) [23], Compensable Workflow Modeling Language (CWML) [16], etc. These languages use the abstract notation of process and control flow in the model to visualize the workflow; however to describe the detailed specification of a process, more is needed. Some workflow systems use XML based languages, while others use general-purpose programming languages (GPLs) such as C++, Java, etc. A control flow consists of two artefacts: i) the flow relation, and ii) branching conditions. When a workflow executes, it follows the flow relation as described in the graphical model and executes the processes (i.e., specifications written in XML or a GPL). During execution, the branching conditions are evaluated to guide the control flow.

We propose $T_\square$ for describing processes and branching conditions: $T_\square$ is a procedural language with declarative feature. For defining the flow relation and visualizing a workflow, existing workflow languages may be used. While $T_\square$ was designed so that it can be used with many workflow systems, we incorporated it into the NOVA Workflow Workbench [13]. The NOVA Workbench consist of several modules, including the NOVA Editor and the NOVA Engine. The NOVA Editor uses a graphical workflow modeling language called CWML. CWML is a block structured workflow modeling language [16] which has compensable components along with common workflow components (e.g., atomic tasks, control flow operators: XOR, OR, AND, etc.). Each atomic task in CWML is associated with a process description file (written in $T_\square$) containing procedures. XOR, OR, and Internal Choice control flow operators require that branching conditions be specified to route the flow; these branching conditions are specified in $T_\square$ as procedures which take branch numbers and return decisions.

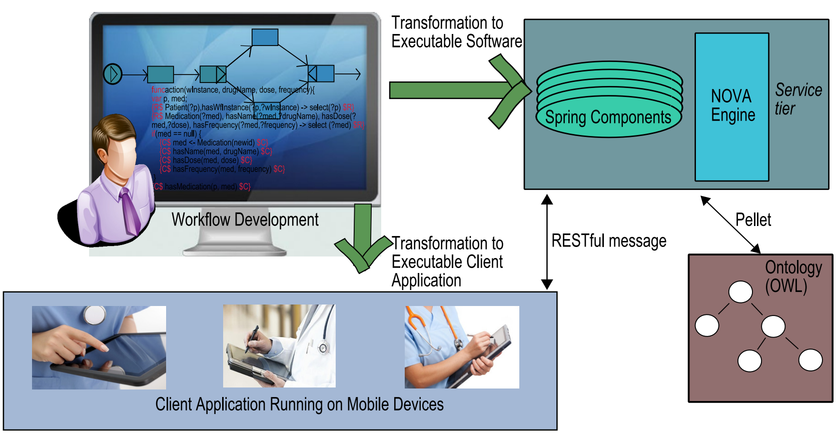


Fig. 1. Overview of the NOVA Workflow Workbench

Fig. 1 shows the overall architecture of the NOVA Workflow Workbench, which incorporates the $T_\square$ editor, developed using Xtext. Specifications written in $T_\square$ are automatically transformed to executable Java programs, using Xtend [2]. In

NOVA Workflow, if a process description file contains a procedure named `view`, the `view` procedure is transformed to a client side Java application. All other procedures are transformed to executable Java server side programs. A `view` method may invoke other procedures; this will initiate an asynchronous data communication to the server by a Web Service (RESTful message). If a process (task) executes, meaning the execution of a procedure named either `action` or `abort` at the server side, the NOVA Workflow engine updates the control flow of the workflow according to the graphical workflow model. As the workflow executes, various operations on the ontology may be required. For example, data may be stored, removed or updated or answeres to queries may be needed. We used Pellet [19], a sound and complete OWL-DL reasoner with extensive support for reasoning. In $T_\Box$, ontology queries are written in the SQWRL (Semantic Query-enhanced Web Rule Language) [15] format. SQWRL is a query language for OWL [5] built on SWRL [4], a rule language which includes a high-level abstract syntax for Horn-like rules. We chose SQWRL because of its simplicity.

3 The $T_\Box$ Language

Workflow management systems often deal with many users and resources. In this section, we present a real life problem to solve with a workflow system, we introduce the $T_\Box$ functionality and syntax and we demonstrate the use of $T_\Box$ to solve the problem.

We designed a graphical workflow model from the 'Guidelines for the management of cancer-related pain in adults' [7]. The guideline suggests the use of Opioids for cancer patients. Opioids are very useful in cancer care to alleviate the severe, chronic, disabling pain but there are common side effects which include nausea and vomiting, drowsiness, itching, dry mouth, miosis, and constipation. The proper use of opioid dosage is important. Fig. 2 shows the tasks

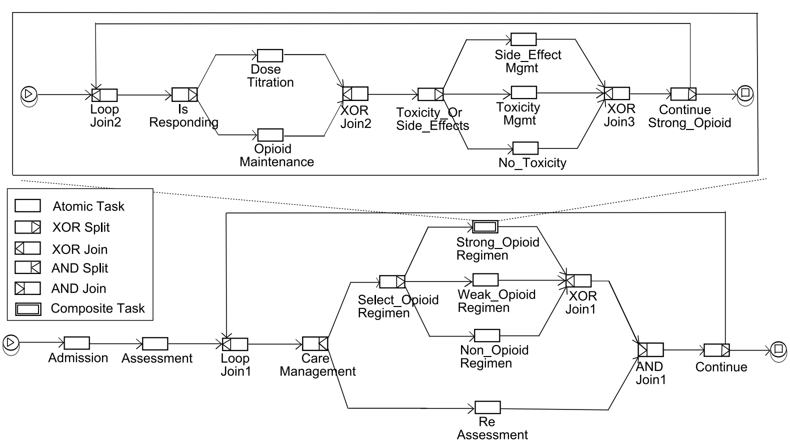


Fig. 2. Pain Management Workflow

for a Pain Management workflow modeled in CWML. A patient is admitted into the system and after that, pain is assessed. The 'Assessment' task assesses all causes of pain, determines pain location, pain intensity, and documents all previous analgesics. A patient is administered his prescribed medicine in one of the 'Strong_Opioid_Regimen', 'Weak_Opioid_Regimen', and 'Non_Opioid_Regimen' tasks; the 'Re_Assessment' task executes concurrently with these tasks. The 'Strong_Opioid_Regimen' is a composite task which is unfolded to a subnet workflow. This subnet workflow deals with any opioid toxicity or side effects found during the treatment procedure.

During the execution of this workflow, caregivers need to interact with the system via user interfaces. The workflow system needs to store and read information to and from a database/ontology. The workflow has decision points (e.g., Select_Opioid_Regimen determines which branch to follow) to guide the control flow. These requirements are articulated using $T_\Box$.

3.1 Writing Procedural Statements

Variables in $T_\Box$ are inferred variables: variable types are determined from their use. Variables in $T_\Box$ may be indexed as array indexes but a declaration of the size is not required. The size is adjusted dynamically at execution time. If no index is used, it refers to the 0^{th} index of a variable. In $T_\Box$, procedures may be invoked by 'call by value' or 'call by reference'. The 'call by reference' of a variable is indicated by a leading '&' . Syntax for Assignment operations, If-Else statements, For-loops, etc., in $T_\Box$ are similar to the C family languages. In $T_\Box$ every procedure returns a value, and return types are not required for procedures. In $T_\Box$ some utility procedures such as `size`, `today`, `currentTime`, `date`, `month`, `year`, `time`, and `tokenize` have been incorporated to deal with string, array, date and time data.

3.2 Querying and Manipulating Ontologies

In many systems, Create, Read, Update, and Delete (CRUD) operations are performed on databases. We use statements with C,R,U,D tags to perform analogous operations on an ontology Abox and refer to the relevant $T_\Box$ statements as OntAssertion, OntRead, OntUpdate, and OntDel statements.

$T_\Box$ allows us to write queries, both for task description and for branch condition, in the SQWRL format. One can perform queries combining Tbox and/or Abox syntax. The Tbox contains concepts and assertions about concepts such as subsumption ($Man \sqsubseteq Person$). The Abox contains role assertions between individuals ($hasChild(John, Mary)$) and membership assertions ($John : Man$). Similar to the 'select' operator of SQWRL, the 'select' operator in $T_\Box$ takes one or more arguments, which must be variables occuring in the body of the query. A particular value may be passed as a query criterion; if a variable is used in an ontology query without a leading question mark,?, then the value is read by the query engine. The following query retrieves all persons from an ontology with a pain intensity that is greater than 5, together with their pain intensities:

```
var  p, pain, v = 5;
{R$ Patient(?p), hasPain(?p, ?pain), greaterThan(?pain, v) →
      select(?p, ?pain) $R}
```

The query engine will populate the variables passed as arguments of the 'select' operator. Selected results may be sorted in ascending or descending order by the 'orderBy' or 'orderByDescending' operators respectively. The following OntAssertion statements create a new 'Patient' individual and inserts a data property for the relation 'hasPain'.

```
var  p;
{C$ p := Patient("Alex") $C}
{C$ hasPain(p, 6) $C}
```

Note that a reference of the newly created Patient individual is assigned to the variable 'p'. An individual may be created with an auto-incremented identity (as below) if in the ontology there exists a data property named 'hasId', where the domain of 'hasId' is 'Thing' and the range is the 'Long' data type.

```
{C$ p := Patient(newid) $C}
```

OntDel statements are used to delete an individual or instance of a relation from an ontology Abox. The following code shows a delete operation of a Patient individual with id=1010.

```
var p, pid = 1010;
{R$ Patient(?p), hasId(?p, pid) → select(?p) $R}
{D$ Patient(p) $D}
```

In this code fragment, a search operation is performed on an ontology for a Patient individual with id=1010 and a reference is retrieved; the Patient individual's reference is then passed as an argument to the delete operation. OntUpdate statements are used to update a data property or object property of an individual. The following code fragment updates the ages of all patients whose birthday is today.

```
var p, P, bDate, Age, age, newAge, cDate = today();
{R$ Patient(?P), hasBirthDate(?P, ?bDate), isEqual(?bDate, cDate),
    hasAge(?P, ?Age) → select(?P, ?Age) $R}
foreach( p in P, age in Age){
    newAge = age + 1;
    {U$ hasAge(p, age => p, newAge) $U}
}
```

3.3 Designing User Interfaces

In many workflow applications, *forms* are used which need many UI view components such as 'Label', 'Text Field', 'Text Area', 'Check box', 'Drop down', 'Date time picker', etc., to capture user input. When a user finishes entering information in a form, the information collected is submitted to the server for processing. Sometimes client side processing on this input is required. The client side processing may involve client server communication for further information, calculation on the provided input, etc. We provide a simple syntax to develop UI forms to deal with these general requirements for client side applications.

To print text or a number in the UI, the `getLabel` procedure may be used. The `getLabel` procedure produces a 'Label' view component in the UI. One can either pass a string literal or a variable as the argument of the `getLabel` procedure. If a variable is passed to the `getLabel` procedure, then the variable is bound to a 'Label' view component. Whenever this variable is updated, the change is reflected in the 'Label'. The following code fragment produces two labels; during execution, the first label will display the text "Workflow Instance:" and the second label will display the number '112'.

```
var wid = 112;
getLabel("WorkflowInstance : ");
getLabel(wid);
```

The `getText` procedure produces a 'Text Field' view component. A 'Text Field' is a common UI component to take user input. The `getText` procedure can take one or two arguments: i) a string to produce a label, and ii) a variable (optional) to display the initial text in a 'Text Field'. A destination variable name after the symbol '>>' is required for the `getText` where the user input is captured. Optionally, some statements (also known as action statements) may be written inside curly braces after the destination variable name of a `getText` procedure. These action statements will be executed when a user finishes her entry into the 'Text Field'. The following code fragment shows an example use of the `getLabel` and `getText` procedures:

```
var hospitalName, displayText = "NoInput";
getText("EnterHospitalName : ") >> hospitalName{
    displayText = "Hospital : " + hospitalName; };
getLabel("EnteredText : ", displayText);
```

This will produce a 'Text Field' where the user will enter text as input; the entered text will be stored in a variable named 'hospitalName'. As soon as the user finishes entering text into the 'Text Field' the action statement (enclosed with curly braces) will execute and sets a value entered by the user to the variable named 'displayText'. Since the variable 'displayText' is bound with a 'Label', when its value changes, the 'Label' view component will be updated and will display the hospital name entered in the 'Text Field'.

The `getInteger` procedure is similar to the `getText` procedure; this also produces a 'Text Field' to take input from the user; the difference is that only numbers are allowed here. The following code fragment gives an example:

```
var basicPay = 14, hourlyPay, totalHr = 0, totalSalary = 0;
getInteger("Hourlypayment : $", basicPay) >> hourlyPay {
    totalSalary = hourlyPay * totalHr; };
getInteger("TotalHourWorked : ") >> totalHr {
    totalSalary = hourlyPay * totalHr; };
getLabel("TotalSalary : $", totalSalary);
```

The first 'Text Field' will display the value of the 'basicPay' variable which is '14'. The user may change it by entering a different number; the entered number will be stored in the variable named 'hourlyPay'. The user enters the total hours worked in the second 'Text Field'. When the user finishes entering numbers in the 'Text Fields', the total salary is calculated and displayed in the UI by a 'Label'.

The `getDouble` procedure is similar to the `getInteger` procedure; the only difference is the user can enter a floating point number in the 'Text Field'. The `getDate` procedure is similar to the `getInteger` procedure but here the user enters a date in a 'Text Field' or in a 'Date Time Picker'. The `getBoolean` procedure takes one argument as input to display a title for a 'Check box' (a view component to select or de-select an item). The user may select or de-select the 'Check box' and a true or false value is assigned to the associated destination variable of a `getBoolean` procedure. If action statements are written for a `getBoolean` procedure, they will be executed after the user selects or de-selects a check box item. The following code will display a 'Check box'.

```
var painCrisis;
getBoolean("PainCrisis") >> painCrisis;
```

The destination variable 'painCrisis' is associated with the 'Check box'. If the 'Check box' is checked, a 'true' value will be set to the variable, otherwise a 'false' value will be set to the variable. Since the 'painCrisis' variable is bound to a 'Check box' view component, if the value is changed from another portion of the procedure, it will be reflected in the UI. This feature may be useful to display a UI form to update existing information. For example, if we want to display a patient's existing Pain Crisis information and allow the user to modify it, then the following code may be used:

```
var painCrisis, id = 1010;
getBoolean("PainCrisis") >> painCrisis;
{R$ Patient(?p), hasId(?p, id), hasPainCrisis(?p, ?painCrisis) →
    select(?painCrisis) $R}
```

The `getOne` procedure is used to select one item from a list of items. This procedure will either display a 'Drop down' view component (if one source variable is provided as argument) or a 'Table' with 'Radio buttons' (if more than one source variable is provided). A destination variable name is required for a `getOne` procedure. Optionally another destination variable name may be specified to store the position of the item selected from the drop down. If action statements are given for a `getOne` procedure, they will be executed as soon as the user selects an item. In the following example, a list of countries is retrieved from an ontology by performing a read operation. A `getOne` procedure is used to display the list of countries in a 'Drop down'. Another `getOne` procedure is used to display a list of provinces in another 'Drop down'. Since the provinces are different from one country to another, on the selection of a country, a further query was performed on the ontology to retrieve related province information; this was done in the action statements. The 'province' variable is bound to the source variable with the second `getOne` procedure; as a result, if a province's information was updated it will be reflected in the 'Drop down'.

```
var c, country, province, selectedProvince;
{R$ Country(?c) → select(?c) $R}
getOne("Country : ", c) >> country {
    {R$ Province(?province), hasCountry(?province, country) →
      select(?province) $R}
};
getOne("Province : ", province) >> selectedProvince;
```

Note that the 'source' variable fills a 'Drop down' view component but if we want to display one item from the items available in the 'Drop down' we may use the 'destination' variable. For instance, in a patient's admission record update form, we want to display the patient's existing province in the 'Drop down'; this can be achieved by assigning the province information to the destination variable.

The `getMultiple` procedure is similar to the `getOne` procedure but here the user may select more than one item from the source variable(s). The `getButton` procedure produces a button in the UI. When a button is pressed, the statements associated with it are executed. For example, if we want to calculate the strength of a given password then a button may be used to do this.

```
var password, result;
getPassword("EnterPassword") >> password;
getButton("CheckPasswordStrength") {
    result = checkPwStrength(password);
    print(result); };
```

When the button "Check Password Strength" is pressed it invokes a procedure named `checkPwStrenth` at the server with 'password' as argument and prints the result in the UI. $T_\square$ provides two procedures to arrange the view components in the UI: `openLayout` and `closeLayout`. The `openLayout` procedure takes an integer parameter which indicates the number of columns. All view components specified after an `openLayout` procedure will follow this arrangement. A `closeLayout` procedure stops putting view components in the layout format begun by an `openLayout` procedure. An `openLayout` procedure can be nested within another `openLayout` procedure allowing the user to implement a complicated table layout structure.

Now we use $T_\Box$ to design a user interface for a task from the Pain Management Workflow in Fig. 2. The code below shows the `view` procedure of the 'Assessment' task.

```
func view(){
 var wInstance, pc;
 . . .//Other variable declaration
 wInstance = getCurrentInstance(); // Get current workflow instance id
 {R$ PainCourse(?pc), hasName(?pc, ?pcName) → select(?pcName)$R}
 . . .// Read other pain assessment information from ontology
 {R$ Drug(?drug) → select(?drug)$R}
 . . .// Read Frequency, Route, Unit information from ontology
 openLayout(2); openLayout(2); // Nested Table Layout
 getMultiple("PainLocation", pLocation) >> painLocation;
 getMultiple("PainTimeOfDay", pTime) >> painTime;
 closeLayout(); openLayout(2);
 getOne("PainDuration", pDuration) >> painDuration;
 . . .// Code to display other view components
 getButton("(+)") {
 drugList[size(drugList)] = selDrug; // Adding a new drug into drugList
 . . .// Add frequency, route, dose information into list
 };
 getButton("(−)"){
 clear(drugList, medPos); // Removing selected item from list (Table)
 . . .// Remove frequency, route, dose information from list
 };
 closeLayout(); openLayout(1);
 getOne("MedicationInformation", drugList "DrugName", freqList
  "Frequency", routeList "Route", unitList "Unit", doseList "Dose")
  >> destDrug, medPos; // Showing medications in a table
 closeLayout();
 // make a submit button to send information to server
 submit(wInstance, painLocation, painTime, painDuration, . . .); }
```

$T_\Box$ generates code for client side applications using the Android [3] platform. The output of this simple 29 lines of code is shown in Fig. 3; this is a screenshot from a Tablet device (operating on an 'Android' operating system). The transformation method automatically produced 1160 lines of Java code, and a few xml configuration files to manage the Android UI, and to deal with network operations. The 'Pain Location', 'Pain Duration', 'Drug Name' etc., information comes from the ontology and is displayed in the UI. The clinician selects a drug name, frequency, route, and unit from 'Drop down' view components and inserts dose in a 'Text Field' and adds them into the 'Medication Information' table by clicking the (+) button. A medication may be removed from the table by selecting it and clicking the (−) button. A patient's opioid regimen is selected after the execution of the 'Assessment' task. We used ontology reasoning to classify medications into opioids. The rules for different opioids (e.g., strong, weak opioid) were incorporated into the ontology. In this way, complex rules of do-

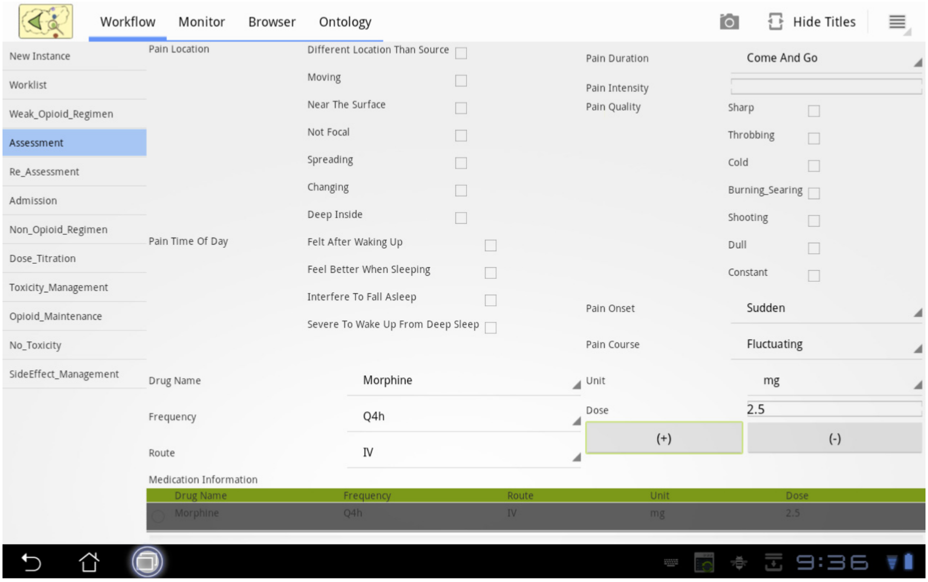


Fig. 3. Assessment form: Output of the `view` procedure of the 'Assessment' task

main knowledge can be effectively handled by using the reasoning power of an ontology.

3.4 Specifying Branch Conditions

During execution of the Pain Management Workflow (Fig. 2) only one outgoing branch of the XOR task 'Select_Opioid_Regimen' is activated. A patient goes into strong opioid regimen if he is currently on a strong opioid or he is on a weak opioid with moderate, severe or unstable pain. A patient goes into the weak opioid regimen if he experiences mild but unstable pain. Otherwise he goes into the non opioid regimen. The code for the task 'Select_Opioid_Regimen' is provided below:

```
xorsplittask Select_Opioid_Regimen;
func getBranchCondition(wInstanceId, brNo){
 var p, pid, pIntensity, pUnderStrong, pUnderWeak, pc, painCourse;
 {R$ Patient(?p), hasWfInstance(?p, wInstanceId), hasId(?p, ?pid),
  hasPainIntensity(?p, ?pIntensity) → select(?p, ?pid, ?pIntensity) $R}
 {R$ PatientUnderStrongOpioid(?pUnderStrong), hasId(pid)
     → select(?pUnderStrong) $R}
 {R$ PatientUnderWeakOpioid(?pUnderWeak), hasId(pid)
     → select(?pUnderWeak) $R}
 {R$ Patient(p), hasPainCourse(p, ?pc), hasName(?pc, ?painCourse)
     → select(?painCourse) $R}
 if(brNo = 1){
  if(pUnderStrong ≠ null || (pUnderWeak ≠ null && pIntensity ≥ 4) ||
```

```
14.    (pUnderWeak ≠ null && painCourse = "GettingWorse") ||
15.    (pIntensity ≥ 4 &&
16.       (painCourse = "Fluctuating" ||painCourse = "Getting Worse")))
17.    return true;}
18.  else if(brNo = 2){
19.   if( (pIntensity ≥ 2 && (painCourse = "Fluctuating" ||
20.      painCourse = "GettingWorse")) || (painCourse = "Getting Worse"))
21.   return true;}
22.  else if(brNo = 3)
23.   return true;
24. return false;}
```

The procedure `getBranchCondition` takes two parameters: a workflow instance id and a branch number and returns true for the branch that should be activated for that instance. Lines 4–11 query the ontology.

4 Related Work

ADEPT2 [17] is an adaptive process management system which supports dynamic change of process schema and definition. The main difference between ADEPT2 and NOVA Workflow is their underlying persistent technology and data structure; ADEPT2 does not support ontologies and the activities in ADEPT2 are written in a GPL. In ADEPT2 web forms are automatically generated from the workflow model although ADEPT2 does not allow action statements for the UI operations. ADEPT2 performs a dynamic validation of process schema change which makes the workflow system consistent. In NOVA Workflow a consistency check is performed whenever any record is inserted into or updated from an ontology Abox.

In [9], the authors presented an evolutionary approach for the model-driven construction of Web service based Web applications on the basis of workflow models which is founded on DSLs and a supporting technical framework. The Workflow DSL is an executable specification language for workflow based Web applications which allows the use of various graphical notations taken from the Business Process Modeling field, e.g., BPMN, Petri Nets, UML activity diagrams etc., as well as custom notations. This model driven approach makes development faster by reusing components but this approach is not ontology based.

In [10], the author worked on an ontology oriented programming and proposed a compiler which produces a traditional object-oriented class library that captures the declarative norms of an ontology. With this approach the developer is required to use a GPL, and the approach is not model driven. In [8], the authors described the Go! language and its use for ontology oriented programming, comparing its expressiveness with OWL. Go! allows knowledge to be represented as a set of labeled theories incrementally constructed using multiple-inheritance. This is related to our work since the authors also proposed a language for building executable ontologies. But the syntax proposed for $T_{\square}$ is simple and abstract, and $T_{\square}$ provides syntax for procedural statements and UI design.

In [6], Baker et. al., surveyed a large number of existing workflow systems and listed their features considering different problem aspects. None, however, follow an ontology based model driven approach as in $T_\square$. Ontology integration and UI form generation may be specified in different ways using existing approaches but they are very easy to specify using $T_\square$.

5 Conclusion

In this paper we presented a new domain specific language, called $T_\square$, for writing specifications for tasks in workflow models. The intension of the language is reflected in the pronounciation "T-Square" – signifying speedy development of tasks (*tasks to the power of 2*). A developer may learn the simple syntax of $T_\square$ and start developing applications without detailed knowledge of the complex API's for Ontologies, Web Services, Android platforms, etc. Code is generated automatically, allowing the developer to fully concentrate on the domain model and system analysis. If there is a change in user requirements, the developer can make the change using $T_\square$ and the NOVA workflow system will automatically update the software accordingly. The output of NOVA Workflow is currently an Android application which runs on mobile devices but different transformations may be applied to generate other applications, for iPad, the Web, etc. End users interact with the client application.

Workflow development with $T_\square$ can benefit from customizing existing and comprehensive ontologies, e.g., SNOMED–CT, ICNP, etc., already in use in healthcare. Since reasoning over a large ontology is time consuming, in future we will work on a bigger case study and deal with the complexity of ontology reasoning where there is a large Abox and complex rules. One approach is to use a relational database and materialize an ontology into a database [22] but this requires further research to speed up the materialization so it can support the necessary frequent updates. This will allow us to retain the benefit of using an ontology to structure and maintain complex relationships. NOVA Workflow with $T_\square$ is currently being evaluated as part of a pilot application with our local health authority. Further $T_\square$ functionalities are under development including a means to safeguard security of (patients') information through task-based access control, and a means to specify dynamic task scheduling.

Acknowledgments. We acknowledge support from the Atlantic Canada Opportunities Agencies, the Natural Sciences and Engineering Research Council of Canada, the Canadian Foundation for Innovation, and the Nova Scotia Research and Innovation Trust. We are grateful for the enthusiastic support of clinicians and administrators from the Guysborough Antigonish Strait Health Authority.

References

1. BPMN: Business Process Model and Notation (BPMN), http://www.omg.org/spec/BPMN/ (last accessed, January 2012)

2. Eclipse xtend, http://www.eclipse.org/xtext/xtend/ (last accessed, January 2012)
3. Google android, http://www.android.com/ (last accessed, January 2012)
4. SWRL, http://www.w3.org/submission/swrl/ (last accessed, January 2012)
5. Web Ontology Language (OWL), http://www.w3.org/2004/owl/ (last accessed, January 2012)
6. Barker, A., van Hemert, J.: Scientific Workflow: A Survey and Research Directions. In: Wyrzykowski, R., Dongarra, J., Karczewski, K., Wasniewski, J. (eds.) PPAM 2007. LNCS, vol. 4967, pp. 746–753. Springer, Heidelberg (2008)
7. Broadfield, L., Banerjee, S., Jewers, H., Pollett, A., Simpson, J.: Guidelines for the Management of Cancer-Related Pain in Adults. Supportive Care Cancer Site Team, Cancer Care Nova Scotia (2005)
8. Clark, K.L., McCabe, F.G.: Ontology oriented programming in Go!. Appl. Intell. 24(3), 189–204 (2006)
9. Freudenstein, P., Nussbaumer, M., Allerding, F., Gaedke, M.: A domain-specific language for the model-driven construction of advanced web-based dialogs. In: Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, pp. 1069–1070. ACM (2008)
10. Goldman, N.M.: Ontology-Oriented Programming: Static Typing for the Inconsistent Programmer. In: Fensel, D., Sycara, K., Mylopoulos, J. (eds.) ISWC 2003. LNCS, vol. 2870, pp. 850–865. Springer, Heidelberg (2003)
11. Jørgensen, J., Lassen, K., van der Aalst, W.: From task descriptions via colored petri nets towards an implementation of a new electronic patient record workflow system. International Journal on Software Tools for Technology Transfer (STTT) 10, 15–28 (2008)
12. MacCaull, W., Jewers, H., Latzel, M.: Using an interdisciplinary approach to develop a knowledge-driven careflow management system for collaborative patient-centred palliative care. In: ACM International Health Informatics Symposium, IHI 2010, Arlington, VA, USA, pp. 507–511. ACM (2010)
13. MacCaull, W., Rabbi, F.: NOVA Workflow: A Workflow Management Tool Targeting Health Services Delivery. In: The Proceedings of 1st International Symposium on Foundations of Health Information Engineering and Systems (FHIES 2011) (2011); Revised version to appear. LNCS (2012)
14. Mernik, M., Heering, J., Sloane, A.M.: When and how to develop domain-specific languages. ACM Comput. Surv. 37(4), 316–344 (2005)
15. O'Connor, M.J., Das, A.K.: SQWRL: A query language for OWL. In: Proceedings of the 5th International Workshop on OWL: Experiences and Directions, OWLED 2009, vol. 529 (2009)
16. Rabbi, F., Wang, H., MacCaull, W.: Compensable WorkFlow Nets. In: Dong, J.S., Zhu, H. (eds.) ICFEM 2010. LNCS, vol. 6447, pp. 122–137. Springer, Heidelberg (2010)
17. Reichert, M., Rinderle, S., Kreher, U., Acker, H., Lauer, M., Dadam, P.: ADEPT2 - next generation process management technology. In: Proceedings Fourth Heidelberg Innovation Forum, Aachen, D.punkt Verlag (April 2007)
18. Schmidt, D.C.: Guest editor's introduction: Model-driven engineering. IEEE Computer 39(2), 25–31 (2006)
19. Sirin, E., Parsia, B., Grau, B., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. Web Semantics: Science, Services and Agents on the World Wide Web 5(2), 51–53 (2007)

20. Staab, S., Studer, R.: Handbook on Ontologies. International Handbooks on Information Systems. Springer (2004)
21. Tetlow, P., Pan, J.Z., Oberle, D., Wallace, E., Uschold, M., Kendall, E.: Ontology driven architectures and potential uses of the semantic web in systems and software engineering. History, W3C, 1–17 (2006)
22. Uddin Faruqui, R.: Scalable reasoning over large ontologies. MSc thesis, St. Francis Xavier University (expected, 2012)
23. van der Aalst, W.M.P., ter Hofstede, A.H.M.: YAWL: yet another workflow language. Information Systems 30(4), 245–275 (2005)

Towards Gesture-Based Process Modeling on Multi-touch Devices

Jens Kolb, Benjamin Rudner, and Manfred Reichert

Institute of Databases and Information Systems
Ulm University, Germany
{jens.kolb,benjamin.rudner,manfred.reichert}@uni-ulm.de
http://www.uni-ulm.de/dbis

Abstract. Contemporary tools for business process modeling use menu-based interfaces for visualizing process models and interacting with them. However, pure menu-based interactions have been optimized for applications running on desktop computers and are limited regarding their use on multi-touch devices. At the same time, the increasing distribution of mobile devices in business life as well as their multi-touch capabilities offer promising perspectives for intuitively defining and adapting business process models. Additionally, multi-touch tables could improve collaborative business process modeling based on natural gestures and interactions. In this paper we present the results of an experiment in which we investigate the way users model business processes with multi-touch devices. Furthermore, a core gesture set is suggested enabling the easy definition and adaption of business process models on these devices. Overall, gesture-based process modeling and multi-touch devices allow for new ways of (collaborative) business process modeling.

Keywords: process modeling, gestures, user-centered process modeling, multi-touch application.

1 Introduction

Multi-touch devices have been increasingly used in companies during the last years and have become more and more useful for the daily work of business people. Multi-touch devices used in companies range from smartphones and tablets to multi-touch tables and walls. Obviously, screen size affects both mobility and application areas. While smartphones are primarily used for mobile communication (e.g., mailing and information retrieval), tablets and multi-touch tables are applied for enabling collaborative tasks (e.g., joint editing of a business document). Consequently, multi-touch devices may be applied to business process modeling as well. In particular, portable multi-touch devices can be used to capture and model business processes while interviewing process participants. Multi-touch devices with larger screens (e.g., multi-touch tables), in turn, can be used to collaboratively model, discuss and change business processes [1]. However, traditional process modeling tools have not been designed with multi-touch

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 280–293, 2012.

devices in mind and do not take their specific properties (e.g., small screen size) and interaction possibilities (e.g., gesture-based interaction) into account [2,3]. In this paper we introduce a set of gestures to easily and intuitively model business processes using multi-touch devices. This gesture set enables process designers to define, visualize and adapt business processes in an intuitive and comprehensible way. The focus of the paper is on the introduction of this core gesture set, and less on a concrete user interface design. The suggested core gesture set is applicable to all screen sizes of multi-touch devices. It has been developed in the context of the proView[1] project [4], which aims at user-centered process management. This includes techniques enabling personalized process model visualizations (e.g., process views [5,6]) as well as intuitive process model representations (e.g., diagrams and trees [7]). Gesture-based modeling complements this work with sophisticated and intuitive concepts for interacting with process designers.

The remainder of this paper is structured as follows: Section 2 gives background information on multi-touch applications and summarizes the abstract modeling operations to be supported by the gesture set. Section 3 presents the results of an experiment we conducted to analyze how users interact with multi-touch devices and which kind of gestures they are using to model and change processes. These results provide the foundation of the core gesture set introduced in Section 4. Finally, Section 5 concludes the paper with a short summary and outlook.

2 Background

This section describes background information needed for the understanding of this paper. Section 2.1 introduces general interaction concepts, which can be used to interact with an application on a multi-touch device. Section 2.2 then lists common characteristics of multi-touch applications. Finally, Section 2.3 presents a core function set required for modeling and adapting business processes.

2.1 Multi-touch Interaction Concepts

Generally, users may interact in different ways with a multi-touch application. First, concepts used in the context of conventional desktop applications may be applied to multi-touch application as well; e.g., *menu-based interactions* use menus and toolbars to provide available functions to users. As an advantage, this concept allows users to easily *explore* the application when searching for a specific function. However, displaying menus and toolbars requires space on the screen, which is limited on devices with small screens (e.g., smartphones). Hence, menu-based interaction should primarily be used for multi-touch applications running on larger screens.

[1] http://www.uni-ulm.de/in/iui-dbis/forschung/projekte/proview-project.html

Gesture-based interaction, in turn, uses gestures for selecting functions of the multi-touch application as well as for interacting with them. Thereby a (multi-touch) *gesture* constitutes a movement of one or more fingers on the screen of the multi-touch device. This movement is then recognized and interpreted by the application. For example, in a slideshow of pictures the *wipe gesture* (i.e., wiping with one finger from right to left) can be used to display the next picture. Generally, gestures can be categorized into *physical gestures* and *symbolic gestures* [8]. Physical gestures manipulate the virtual objects on the screen directly, e.g., by dragging a virtual element on the screen. Symbolic gestures, in turn, are related to the function to be executed (e.g., drawing a plus to add an object). As opposed to menu-based interactions, gestures do not require any space on the screen in order to display menus or other graphical elements. However, as a barrier for their application, users do not always know which functions are supported by a multi-touch application and with which gestures they can be selected. The latter is especially crucial for novices and unexperienced users of the multi-touch application. Usually, this problem is addressed through a tutorial provided the first time the user starts the application. Finally, a *hybrid interaction* based on both menu- and gesture-based interactions can be realized. For example, a menu bar may be displayed at the top or bottom of the screen offering the most important functions. Additionally, well-known gestures may be supported, e.g., the above mentioned wipe gesture.

2.2 Characteristics of Multi-touch Applications

Besides their interaction concepts multi-touch applications—or to be more precise their multi-touch capabilities—can be described through six properties.

C1 (Screen Occlusion): The user interface design of a multi-touch application should consider that users directly interact with the application, but might cover screen areas with their hand [9]. For example, dragging an object from the top-left corner of a tablet to its bottom-right corner will be a difficult task if large parts of the screen are covered by the user's hand.

C2 (Handling Precision): Conventional desktop applications, require a mouse pointer to interact with them. This pointer scales with the screen resolution in order to ensure high precision of the interactions. In multi-touch applications, in turn, the 'interaction device' is the user's finger [10]. As opposed to the mouse pointer, a finger does not scale up or down when changing screen sizes. Accordingly, the handling precision of a multi-touch application changes with the size of an object the user has to touch. Studies have shown that an object should have a size of 11.52 mm or more to have a hit probability of at least 95% [11]. Regarding multi-touch process modeling, it will be crucial to achieve a high handling precision.

C3 (Fatigue of Extremities): When using a computer mouse, usually, the hand does not move a lot. Typically, the arm lays on the table and the mouse is moved with the fingers to reach the corners of the screen with the mouse pointer. Interacting with multi-touch applications, in turn, frequently requires the movement of the whole arm. Further, the arm does not lay on a table, but has to be hovered in the air during the interaction with the multi-touch application [12]. Obviously, the degree of movement strongly depends on the screen size.

C4 (Number of Supported Fingers): Compared to conventional desktop applications with a mouse pointer, a multi-touch application typically supports more than one touch point. Generally, the maximum of touch points an application supports is limited through the underlying hardware and operation system. For example, Apples iPad distinguishes between 11 touch points (i.e., fingers) [13]. While this number is sufficient for single user applications, multi-user applications should be able to even support more touch points, e.g., in order to allow for the concurrent modeling of processes on a multi-touch table.

C5 (Number of Concurrent Users): Multi-touch applications may be concurrently used by multiple users. Obviously, the degree of concurrency is limited by the screen size. While concurrent interactions with a tablet would effect each other, concurrent use of an application on a multi-touch table (e.g., joint editing or modeling) might improve the way of working [14]. Especially in the context of concurrent process modeling, this should be exploited, particularly in process management projects involving multiple modelers [15].

C6 (Usage of Common Interaction Concepts): As known from conventional desktop applications, there are interaction concepts representing de-facto standards. Examples include the menu bar on the top of the application window or *File* as first entry of such a menu bar. Such recurrent patterns help users to intuitively interact with applications. The same applies to multi-touch applications and especially gesture-based interactions; e.g., the *pinch gesture* is typically used for zooming. However, there are only few gestures that have been used in a consistent manner so far. Therefore, [16] suggests a gesture dictionary with the intention that different applications show similar behavior in connection with a specific gesture.

The introduced characteristics of multi-touch devices show their wide range of possible applications. Obviously, these characteristics should be taken into account when designing gestures for business process modeling.

2.3 Required Functions for Modeling Business Processes

This section introduces a core function set required to model and change processes. This function set is derived from the change patterns presented in [17,18]. In the context of this paper we assume that process models are well-structured and are based on an activity-centric modeling language. Furthermore, a process model is displayed to process designers by using advanced layout algorithms, i.e., the process designer cannot position single process elements arbitrarily on the screen. Finally, an initial process model consists of a start and an end node connected through a control-flow edge. Based on modeling functions F1-F8, as introduced in the following, the (multi-touch) application ensures that the elements of a process model are connected.

Function *F1 (Insert Activity)* inserts a single activity into a process model between two existing nodes. *F2 (Insert Surrounding Block)* encloses a process fragment with an XOR (AND) block, i.e., an XOR (AND) split with corresponding XOR (AND) join gateway. The enclosed process fragment may be also "empty" leading to the creation of an XOR (AND) block having one "empty" branch. Function *F3 (Insert Branch)* extends an existing XOR (AND) block with an empty branch between the splitting and joining gateway.

Function *F4 (Renaming Element)* changes the name of a process element, e.g., an activity or data element. To delete process elements, function *F5 (Delete Element)* can be applied. The process element may be an activity, a data element, or a gateway. Comparable to F1, function *F6 (Insert Data Element)* adds a new data element to the process model. This data element, in turn, can be connected to activities using function *F7 (Insert Data Edge)*. Finally, function *F8 (Extract/Inline Sub-process)* extracts a sub-process based on a selected set of activities or inlines this sub-process [19]. This function is especially required when modeling complex process hierarchies. An overview of the core function set is given in Table 1.

Table 1. Core Function Set

Core Function Set	
F1 Insert Activity	F5 Delete Element
F2 Insert Surrounding Block	F6 Insert Data Element
F3 Insert Branch	F7 Insert Data Edge
F4 Renaming Element	F8 Extract/Inline Sub-process

In this paper, we restrict ourselves to this core function set, which provides elementary functions required to model and adapt processes. Obviously, additional functions are required for a sophisticated process modeling tool (e.g., to specify attributes or to undo/redo modeling steps).

3 An Experiment on Multi-touch Process Modeling

Experimental research in the context of business process management has already shown promising results regarding user-centric process support [20,21,22,23]. In our research, we conducted, a user experiment on multi-touch process modeling before developing a corresponding gesture set. Using the *Goal Definition Template* as presented in [24], the goal of the experiment is defined as follows:

Analyze	*multi-touch process modeling interaction*
for the purpose of	*evaluating*
with respect to their	*intuitive usage*
from the point of view of	*the researchers and developers*
in the context of	*students and research assistants.*

Taking this goal definition, our experiment evaluates multi-touch interactions applied by users—or to be more precise by students and research assistants—when modeling and changing processes on multi-touch devices. The results of this experiment are then used to develop a consistent set of modeling gestures covering the core process modeling functions introduced in Section 2.3. Based on this goal definition, the following hypothesis is derived:

Users intuitively use multi-touch gestures when modeling and changing process models on multi-touch devices.

Section 3.1 refines the goal and explains the experiment setting. Section 3.2 discusses the results of the experiment.

3.1 Experiment Setting

The experiment is designed as a *multi-test within object study* [24], i.e., having two groups of participants and one object. *Students* (as first group) do not have a broad background in process modeling and process management, whereas the second group consists of research assistants being more experienced with process modeling. Furthermore, the experiment is divided into two parts. In the first one, demographic and background information of the participants is collected (i.e., gender, experience with multi-touch devices, profession). In the second part of the experiment, each participant has to modify an existing process model (i.e., object) in eight different steps. Each step includes a description explaining to participants what they have to do in order to perform the respective process modeling task (cf. Section 2.3). For example, step "Think of a way to create a new activity between *Make Up Package* and the *XOR Branch*" can be accomplished using core function *F1 (Insert Activity)*.

The response variable of our experiment is the *interaction category* used to perform the required change of the process model. Therefore, for each step the variable can have one of the following values: *picture-based*, *gesture-based*, or *menu-based interaction*.

As instrumentation of the experiment, an Apple iPad and the multi-touch drawing application Doodle Buddy [25] is used. For each step, this application displays an image to the participant. On the image, a process model is shown serving as starting point for the model change to be conducted in the respective step. Furthermore, the aforementioned textual explanation is displayed in this context. Finger movements of participants are captured through lines overlaying the image (cf. Figure 1). Additionally, the gesture is captured through a video camera. The latter records the screen of the device as well as the hands of the participant and their movements. Furthermore, participants are asked to *think aloud* about what they are doing and what they are thinking about [26]. This information is used for classifying and interpreting results afterwards (cf. Section 3.2). Furthermore, the supervisor of the experiment stays in the same room as the participant, motivates him or her to *think aloud*, and provides assistance in case of emerging questions.

3.2 Analysis and Results

26 participants attended the experiment. Table 2 summarizes background information on them. It is worth mentioning that the number of participants being experienced with multi-touch devices (e.g., smartphones) and participants not using multi-touch devices is almost equal.

Table 2. Participants' Background

	Gender	**Profession**	**Handedness**	**Multi-Touch Experience**
26 Participants	19 Male 7 Female	17 Students 9 Res. Assist.	25 Right 1 Left	14 Yes 12 No

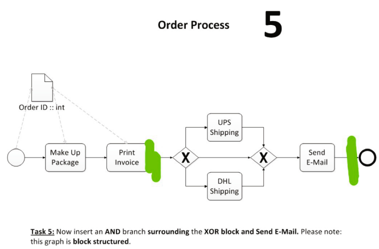
We classify the interactions a participant applies in the context of a concrete modeling step into three *interaction categories*. Note that this classification is accomplished by two persons and is therefore a subjective measurement. As basis of this measurement, we choose the drawing overlaying the image of the respective modeling step as well as the video capturing the actual movement of the participant's hand.

The first category groups *gesture-based interactions*. This kind of interaction uses simple movements on the multi-touch screen changing the process model (as physical gestures described in Section 2.1). Figure 1a exemplarily shows a gesture-based interaction applied to insert an XOR block surrounding a particular process fragment in a process model. In this scenario, the participant moves two fingers simultaneously up and down in order to insert the surrounding block.

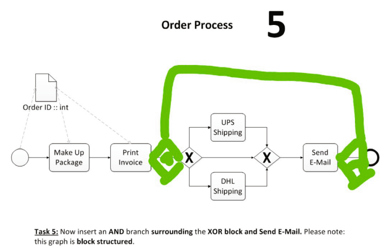
The second category comprises *picture-based interactions*, i.e., all interactions drawing a realistic representation of the final result expected from the application of the respective modeling function (as symbolic gestures, cf. Section 2.1). Figure 1b shows an example of this category.

The third category captures all interactions presuming the presence of a menu bar or context menu. Figure 1c exemplarily shows the result of a participant who is drawing a toolbar at the top of the screen and is dragging & dropping the required process elements on the depicted process model. This category, therefore, covers *menu-based interactions*.

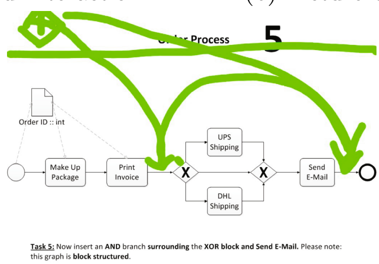
Figure 2 summarizes the results of the experiment and visualizes the distribution of the interaction categories. On the x-axis, each step of the experiment



(a) Gesture-based Interaction



(b) Picture-based Interaction



(c) Menu-based Interaction

Fig. 1. Interaction Categories of the Experiment

is represented through its corresponding function. The y-axis, in turn, shows the number of participants using interactions from a specific category.

Obviously, there is no predominant interaction concept covering all process modeling functions. However, for certain functions one can observe a clear preference for a specific interaction concept. Function *F8 (Extract/Inline Sub-process)*, for example, is applied by 73% of the participants by using gesture-based interaction. For other functions (e.g., *F2 (Insert Surrounding Block)*), however, no dominating interaction can be identified. Hence, multiple interaction concepts should be provided to optimally support different user groups and users in applying respective functions.

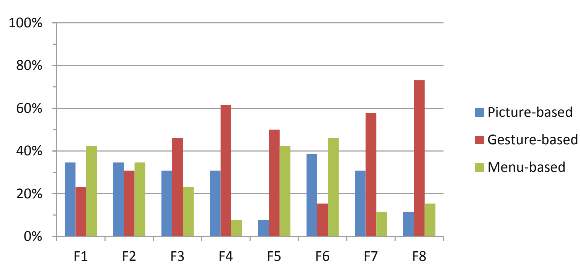
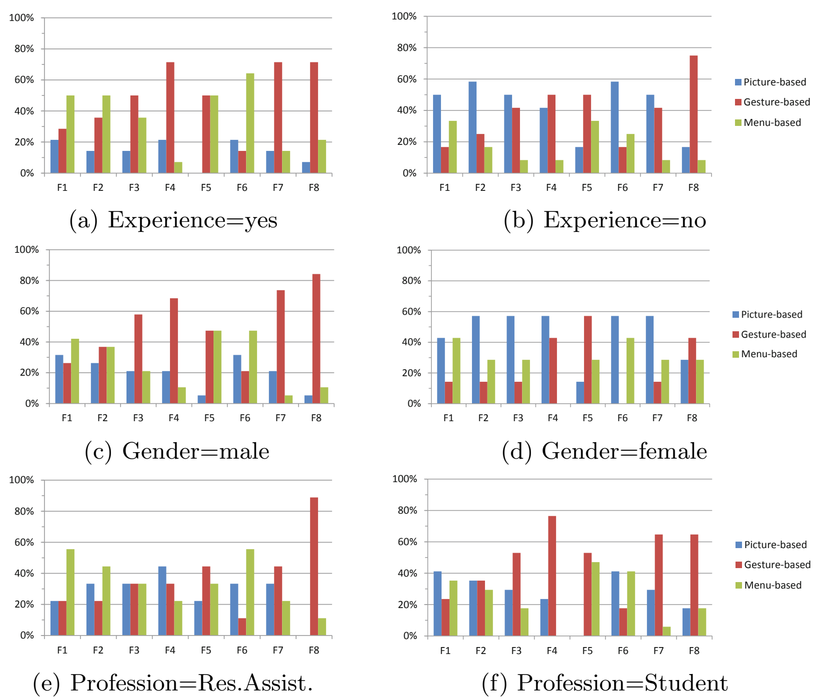


Fig. 2. Distribution of Interaction Categories

When drilling down the results, some insights into differences of the interactions applied by the two groups of participants can be identified. Analyzing the influence of multi-touch experience, for example, shows that experienced participants use gesture-based interactions more often than picture- and menu-based interactions (cf. Figure 3a). By contrast, unexperienced participants more often tend to predominantly use picture-based interactions, but do hardly use menus (cf. Figure 3b). Probably, this can be explained by the fact that unexperienced participants are unaware of common interaction concepts as provided by multi-touch devices (cf. Section 2.2).

Comparing the results of the two gender groups (cf. Figures 3c+3d), female participants predominantly use picture-based interactions. Moreover, none of the female participants uses gesture-based interactions when applying function *F6 (Insert Data Element)*. Male participants, in turn, show a clear preference for gesture-based interactions.

Finally, comparing the results of the professions, participants with low experience on process modeling (i.e., students) slightly prefer gesture-based interactions (cf. Figure 3f). Opposed to this, participants having modeling experience (i.e., research assistants) do not clearly prefer a specific interaction except for function F8 (cf. Figure 3e). Overall, here no clear distinction can be made.

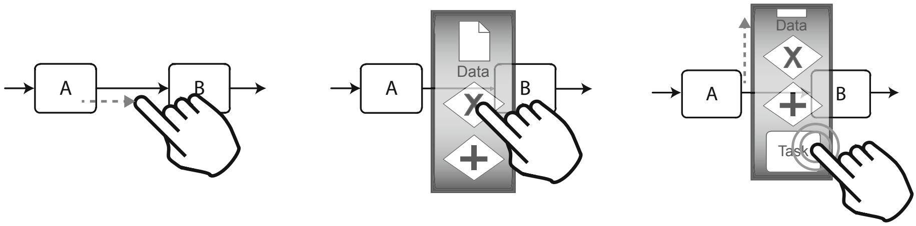


(a) Experience=yes (b) Experience=no

(c) Gender=male (d) Gender=female

(e) Profession=Res.Assist. (f) Profession=Student

Fig. 3. Experiment Results



(a) Wipe to Right (b) Scroll the Menu (c) Tap on Element

Fig. 4. Gesture for Inserting New Elements

Note that the results of our experiment are only to a certain degree representative due to the rather low number of participants and the chosen experiment setting. Nevertheless, it can be concluded that process modeling experience does not influence the way how processes are modeled on multi-touch devices. By contrast, being experienced with multi-touch devices affects the way of modeling processes on them. This can be explained by the non-availability of standardized gestures on multi-touch devices (cf. Section 2). Therefore, when developing a multi-touch application, its target group should be taken into account as well; e.g., it might be relevant to know whether members of the target group use their own devices or multi-touch devices are handed out to them.

In the following, we focus on end-users being experienced with multi-touch devices, and propose a core gesture set enabling intuitive process modeling.

4 A Core Gesture Set for Process Modeling

Taking into account the results of our experiment, we propose a core gesture set for modeling and adapting processes. In this context, we have to cope with the existing trade-off between common interaction concepts provided by multi-touch devices and the results of our experiment. Our goal is to provide an appropriate gesture set suited for modeling and adapting processes on multi-touch devices.

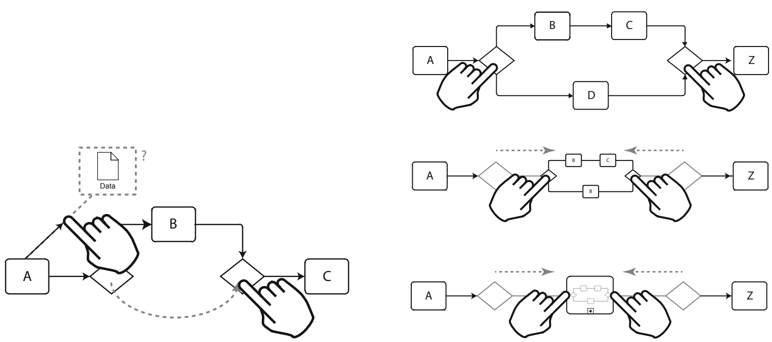
4.1 Core Gesture Set

Our experiment shows that users prefer a menu-based interaction when adding new elements to a process model (cf. Figure 2). Therefore our core gesture set includes a slider menu enabling the insertion of activities, data elements, and surrounding control-flow blocks (cf. functions F1, F2, F6). To trigger the slider menu, users wipe with their finger from an existing process element right to insert the new element at the respective position (cf. Figure 4a). Following this, the slider menu appears at the position where the user releases his finger from the surface of the multi-touch device. Through up and down movements in the slider menu, the required process element can be chosen. For example, to insert an element, the user taps on the respective icon in the slider menu (cf. Figure 4c);

afterwards the slider menu disappears and the element is inserted. Obviously, when inserting a surrounding control-flow block, a second position needs to be selected for adding the corresponding join gateway. For this purpose, the process model editor shows valid positions in the process model where the joining gateway may be inserted; the user then chooses a position by tapping on it. To provide more sophisticated user support when inserting a control-flow block that surrounds an existing process fragment, abstractions in terms of *process views* are useful, i.e., abstracting the process model to a simpler one only comprising those activities being relevant for the insertion of a join gateway [5].

The ordering of the process elements depicted in the slider menu can be optimized by considering their usage frequency. For example, inserting an activity might be more often required compared to the insertion of a surrounding control-flow block. Therefore, the respective function should be positioned on a "central" position in the slider menu that can be quickly accessed.

Another core gesture allows inserting new edges into a process model (cf. Figure 5a). Since we presume a block-structured modeling approach with fixed layout, in general, only two types of edges need to be "manually" added: data edges and edges connecting split and join gateways to add "empty" branches. For example, the user has to draw a line with his finger from the splitting to the joining gateway in order to insert an "empty" branch between them (i.e., F3).



(a) Insert Edges (b) Extract/Inline Sub-process

Fig. 5. Gesture to Insert Data Edges and Extract/Inline Sub-processes

To insert a data edge, in turn, the user has to draw a line between the activity and the respective data element or vice versa (i.e., F7). The direction of the data edge indicates whether it represents a read or write access of the activity on the data element. When inserting a data edge, the gesture set supports users by suggesting a target element (see the left of Figure 5a). If the target element is the desired one the user can lift his finger and the edge will be automatically completed and inserted. Note that this gesture complies with our experiment results since for both functions (i.e., F3 and F7) the majority of participants prefer gesture-based interactions.

To extract or inline a sub-process (i.e., function F8), a two-handed gesture is introduced. The user pushes the start and end element of the process fragment to be extracted into a sub-process activity (Figure 5b). While pushing the elements towards each other, the movement needs to be animated to give direct feedback to the user. After completing the gesture, the new sub-process model has to be linked to the respective sub-process activity. The resulting sub-process can then be displayed through a double tap gesture. To inline a sub-process, in turn, the user pulls the sides of the activity representing the sub-process apart. When pulling the sides of this activity apart, the sub-process appears and is inlined into the process model again. Note that this gesture also matches the results of our experiment in which a majority of 73% of the participants use gesture-based interaction in the context of function F8.

Though 62% of the participants of the experiment use gesture-based interaction to rename a process element (i.e., free-hand writing on the screen), most mobile applications provide an on-screen keyboard for inserting or modifying text on multi-touch devices. The keyboard is hidden most of the time to save screen space and is displayed only when the user wants to add or modify text. To comply with common interaction (de-facto) standards on multi-touch devices, in our gesture set, function *F4 (Renaming Element)* can be activated through tapping on a process element. Afterwards the keyboard appears and the user may modify the text. After confirming the modified text, the keyboard disappears.

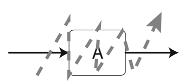


Fig. 6. Delete Element Gesture

Finally, the majority of participants delete process elements using gesture-based interactions. For realizing this function, a gesture crossing out the element to be deleted is suggested (cf. Figure 6). Generally, different variants for realizing this gesture could be envisioned (e.g., drawing two crossing lines or one line from bottom-left to top-right). Regarding our core gesture set, the user draws a connected zigzag-line on the respective element. Generally, this is more accurate regarding recognition compared to the drawing of two separate lines.

5 Summary and Outlook

In this paper, we presented the results of a first experiment we performed to investigate how users model processes on multi-touch devices in a natural and intuitive way. Experiment results indicate that gestures are useful for interacting with process modeling functions, but that for some functions menus are preferred when editing process models. Based on our experiment results as well as common interaction concepts of multi-touch applications a core gesture set has been suggested. This gesture set provides the basis for process modeling tools offering new ways of gesture-based process modeling and providing a high degree of usability. Note that it is no good idea to migrate existing modeling tools, which were designed for desktop computers, straight forward to multi-touch devices since this would affect the usability. The gesture set developed is independent from a concrete device type (e.g., table, tablet or smartphone), but has to be extended with gestures for supportive functions (e.g., undo/redo, copy, paste) in order to use it in a production environment.

As next step a proof-of-concept prototype will be developed. Using this modeling application, further experiments will be conducted to evaluate usability of the developed core gesture set and to elaborate emerging opportunities for modeling processes on multi-touch devices.

References

1. Frisch, M., Heydekorn, J., Dachselt, R.: Diagram Editing on Interactive Displays Using Multi-touch and Pen Gestures. In: Goel, A.K., Jamnik, M., Narayanan, N.H. (eds.) Diagrams 2010. LNCS, vol. 6170, pp. 182–196. Springer, Heidelberg (2010)
2. Gong, J., Tarasewich, P.: Guidelines for Handheld Mobile Device Interface Design. In: Proc. DSI 2004 Annual Meeting, pp. 3751–3756 (2004)
3. Wang, X., Ghanam, Y., Maurer, F.: From Desktop to Tabletop: Migrating the User Interface of AgilePlanner. In: Forbrig, P., Paternò, F. (eds.) HCSE/TAMODIA 2008. LNCS, vol. 5247, pp. 263–270. Springer, Heidelberg (2008)
4. Rudner, B.: Fortgeschrittene Konzepte der Prozessmodellierung durch den Einsatz von Multi-Touch-Gesten. Bachelor thesis, Ulm University (2011) (in German)
5. Reichert, M., Kolb, J., Bobrik, R., Bauer, T.: Enabling Personalized Visualization of Large Business Processes through Parameterizable Views. In: Proc. 26th Symposium on Applied Computing, SAC 2012, Riva del Garda, Trento, Italy (2012)
6. Bobrik, R., Bauer, T., Reichert, M.: Proviado – Personalized and Configurable Visualizations of Business Processes. In: Bauknecht, K., Pröll, B., Werthner, H. (eds.) EC-Web 2006. LNCS, vol. 4082, pp. 61–71. Springer, Heidelberg (2006)
7. Kolb, J., Reichert, M., Weber, B.: Using Concurrent Task Trees for Stakeholder-centered Modeling and Visualization of Business Processes. In: Oppl, S., Fleischmann, A. (eds.) S-BPM ONE 2012. CCIS, vol. 284, pp. 237–251. Springer, Heidelberg (2012)
8. Wobbrock, J.O., Morris, M.R., Wilson, A.D.: User-defined Gestures for Surface Computing. In: Proc. 27th Int'l Conf. on Human Factors in Computing Systems, CHI 2009, New York, USA, vol. 1083 (2009)
9. Brandl, P., Leitner, J., Seifried, T., Haller, M., Doray, B., To, P.: Occlusion-Aware Menu Design for Digital Tabletops. In: Proc. 27th Int'l Conf. on Human Factors in Computing Systems, CHI EA 2009 (2009)
10. Benko, H., Wilson, A.D., Baudisch, P.: Precise Selection Techniques for Multi-Touch Screens. In: Proc. SIGCHI Conf. on Human Factors in Computing Systems, CHI 2006, pp. 1263–1272 (2006)
11. Wang, F., Ren, X.: Empirical Evaluation for Finger Input Properties in Multi-Touch Interaction. In: Proc. 27th Int'l Conf. on Human Factors in Computing Systems, CHI 2009, pp. 1063–1072 (2009)

12. Yee, W.: Potential Limitations of Multi-touch Gesture Vocabulary: Differentiation, Adoption, Fatigue. In: Jacko, J.A. (ed.) HCI International 2009, Part II. LNCS, vol. 5611, pp. 291–300. Springer, Heidelberg (2009)
13. Gemmell, M.: iPad Multi-Touch (2010), http://mattgemmell.com/2010/05/09/ipad-multi-touch/ (last visited: February 06, 2012)
14. Hornecker, E., Marshall, P., Dalton, N.S., Rogers, Y.: Collaboration and Interference: Awareness with Mice or Touch Input. In: Proc. 2008 ACM Conf. on Computer Supported Cooperative Work, CSCW 2008, pp. 167–176 (2008)
15. Edelman, J., Grosskopf, A., Weske, M., Leifer, L.: Tangible Business Approach Process Modeling: A New Approach. In: Proc. Int'l. Conf. on Engineering Design, ICED 2009, pp. 489–500 (2009)
16. Elias, J.G., Westerman, W.C., Haggerty, M.M.: Multi-Touch Gesture Dictionary. US Patent & Trademark Office, US7840912 (2007)
17. Weber, B., Reichert, M., Rinderle, S.: Change Patterns and Change Support Features - Enhancing Flexibility in Process-Aware Information Systems. Data & Knowledge Engineering 66(3), 438–466 (2008)
18. Rinderle-Ma, S., Reichert, M., Weber, B.: On the Formal Semantics of Change Patterns in Process-Aware Information Systems. In: Li, Q., Spaccapietra, S., Yu, E., Olivé, A. (eds.) ER 2008. LNCS, vol. 5231, pp. 279–293. Springer, Heidelberg (2008)
19. Weber, B., Reichert, M., Mendling, J., Reijers, H.A.: Refactoring Large Process Model Repositories. Computers in Industry 62(5), 467–486 (2011)
20. Weber, B., Mutschler, B., Reichert, M.: Investigating the Effort of Using Business Process Management Technology: Results from a Controlled Experiment. Science of Computer Programming 75(5), 292–310 (2010)
21. Recker, J., Dreiling, A.: Does It Matter Which Process Modelling Language We Teach or Use? An Experimental Study on Understanding Process Modelling Languages without Formal Education. In: Proc. 18th Australasian Conference on Information Systems, Toowoomba, Australia, pp. 356–366 (2007)
22. Recker, J., Safrudin, N., Rosemann, M.: How Novices Model Business Processes. In: Hull, R., Mendling, J., Tai, S. (eds.) BPM 2010. LNCS, vol. 6336, pp. 29–44. Springer, Heidelberg (2010)
23. Mutschler, B., Weber, B., Reichert, M.: Workflow Management versus Case Handling: Results from a Controlled Software Experiment. In: Proc. 23rd Annual ACM Symposium on Applied Computing, SAC 2008, Special Track on Coordination Models, Languages and Architectures, pp. 82–89 (2008)
24. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslen, A.: Experimentation in Software Engineering - An Introduction. Kluwer Academic Publishers (2000)
25. Pinger Inc.: Doodle Buddy (last visited: February 06, 2012), http://itunes.apple.com/de/app/doodle-buddy/id313232441?mt=8
26. Jaspers, M.W., Steen, T., van den Bos, C., Geenen, M.: The Think Aloud Method: A Guide to User Interface Design. Int'l. Journal of Medical Informatics 73(11), 781–795 (2004)

Individual Creativity in Designing Business Processes

Kathrin Figl[1] and Barbara Weber[2]

[1] Vienna University of Economics and Business (WU Vienna),
Augasse 2-6, 1090 Vienna, Austria,
kathrin.figl@wu.ac.at
http://nm.wu-wien.ac.at/nm/figl

[2] University of Innsbruck, Technikerstrasse 21a, 6020 Innsbruck, Austria

Abstract. Designing business processes in a creative way is an important requirement for implementing process-aware information systems. In this article we investigate how process modeling competence and individual creativity style and capacity influence creativity in a business process redesign task. We explore these relationships with a laboratory experiment with 48 business students. Our preliminary results showed that process modeling competence is positively associated with the creative quality of a business process redesign, while individual creativity style and capacity measured by well-known creativity inventories seem to be less relevant. The findings underline the importance of training in process modeling to enable employees to realize their full creative potential when redesigning process models in process improvement projects.

Keywords: creativity, business process redesign, process modeling.

1 Introduction

Process models define the organizational procedures intended to be supported by an information system. For this reason, up-to-date and innovative process models form the basis for adequate process-aware information systems leading to organizational success.

According to the Gartner Group [9] process improvement is one of most important priorities for top management. Redesign ideas challenging existing business processes play an ever increasing role. They can lead to radical change or to continuous improvement of processes and better business performance in the long run. In that context, visual process models can help to identify process weaknesses as well as possible improvement opportunities. Visual modeling of TO-BE processes helps to imagine how changes would affect the entire business process. Moreover, in a modeling tool it is easily possible to model and subsequently analyze a variety of innovative changes and improvement opportunities for a process.

For creating novel and valuable ideas for process redesign, creative ability of the employees working with the process models is needed. According to Mumford

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 294–306, 2012.

[18, p. 1] "creativity and innovation are key requirements for the growth and adaptation of organizations". Companies have to identify their creative potential to enable business innovation. Identifying individual creative potential, in turn, is relevant for assembling teams and deciding who should be part of a process redesign team. Additionally, creativity training can be of relevance for teams working on processes. Nowadays, only few firms encourage creative processes or offer creativity training beyond financial incentives [3], which may even hamper creative efforts by reducing intrinsic efforts. In this article we focus on creativity on the individual level in the task context of business process improvement.

Yet, while creativity has been recognized as important requirement for business success, surprisingly little research has been published on creativity in information systems journals. Creativity is still an under-researched area in the information systems discipline and only a limited number of creativity related research questions have already been investigated [23]. A recent literature review on creativity in the information systems field [23] illustrates which research questions have been addressed so far and demonstrates that creativity is finally receiving increasing attention in the field. To give some examples - a large stream of research has for instance dealt with the design and evaluation of creativity support systems (see e.g. [17]); others have analyzed which creativity improvement techniques are appropriate for solving problems related to information systems [4] or discussed the concept and management of creativity in business process management, for instance, in creative intensive fields [25, 24].

In our work we continue along the growing interest in creativity in the information systems field. Given the high practical relevance of creative process redesign, the aim of this study is to explore the relationship between individual creativity and business process redesign. More specifically, we address the research question how creative personality style and creative capacity influence creativity in a process redesign task.

This paper is structured as follows: First, we provide a background to business process redesign and individual factors for creativity. Next, we describe the research model underlying our empirical study. Then we explain the experimental design, before the next section presents our creativity analysis and an examination of the results. The final sections discuss our results and possible limitations of the study.

2 Theoretical Background

2.1 Business Process Redesign

Business process reengineering and continuous process improvement constitute two different ends on a continuum of business process redesign. *Business process reengineering* refers to "fundamental rethinking and radical redesign of business processes to achieve dramatic improvements in critical, contemporary measures of performance, such as cost, quality, service and speed" [10, p. 32]. Thereby, business process reengineering largely ignores the existing process and in the most extreme case follows a *clean sheet* approach creating the redesigned process from scratch. *Continuous process improvement*, in turn, typically takes existing processes as a starting point and gradually improves them in an incremental manner [5]. While both approaches differ in their focus, both aim at the redesign of existing business processes and lead to a transformation of the original process model (i.e., AS-IS process) into a redesigned version (i.e., TO-BE process) improving one or more performance measures. To achieve this transformation different redesign patterns can be used as for instance cutting non-value adding tasks, changing the order of tasks or automating tasks [22].

2.2 Creativity

Creativity and creative problem-solving form the basis for business innovation and process redesign [14]. Similarly to other design activities, there is no single solution and no clear path to a solution in a process redesign task. A creative solution is defined as being both original and novel as well as relevant and valuable in the specific context [1]. In a business setting novelty and uniqueness of ideas are especially desirable because such ideas can lead to competitive advantage towards competitors [28]. When researching creativity three main aspects are relevant: the *creative person*, the *creative process* and the *creative product* [1]. In literature, researchers have also applied a fourth factor for understanding creativity in the so-called 4-Ps model: *the creative press or environment* [4]. In our study we intend to include three aspects and examine how creative personality (*creative person*) influences the solutions to a process redesign-task (*creative product*) using a modeling tool (*creative process*), the fourth factor - *creative environment* is held constant due to the laboratory setting.

2.3 Individual Influence Factors for Creativity

Although it is possible to increase individual creativity by training to some extent, some individuals are simply more creative than others [1]. Research on individual factors for creativity has revealed that besides task motivation, domain-related skills and creativity relevant skills and abilities influence the individual's capacity for creative action [6, 1]. According to Amabile's component theory of creative performance [1, p. 367] "each component is necessary, and not one is sufficient for creativity in and of itself". Concerning domain-related skills, formal education, knowledge about a domain and technical skills are relevant [1]. Additionally, prior research has taken personality traits and characteristics as essential influence factors for idea generation into account. One of the most influential personality models in the context of creativity is the Adaption-Innovation theory by Kirton [11]. While adaptors tend to "doing things better", innovators are likely to be "doing things differently" [11]. Adaptors work out few, but realizable solutions, they pay attention to details and work out the solution incrementally with well-known techniques [7]. In contrast, innovators propose more, but less realistic solutions and might ignore rules and existing paradigms. This distinction directly relates to the continuum between continuous, incremental improvement [5] and radical redesign of processes [10].

3 Research Model and Hypotheses

Having laid out the relevant theoretical foundation related to creativity and business process redesign, we will now draw several propositions to suggest which factors will influence the creativity of a business process redesign. Concerning person-related characteristics creative style and creative capacity are two independent dimensions [12] which are both relevant for solutions to process improvement tasks. Besides person-related characteristics, domain-relevant skills are another factor contributing to the creative process [1]. We summarize our expectations about relevant influence factors in light of the theoretical considerations in the research model shown in Figure 1.

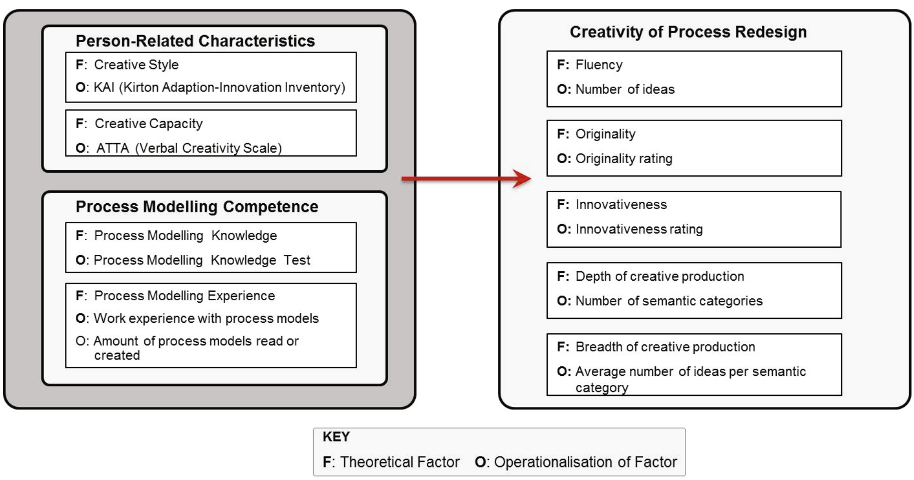


Fig. 1. Research Model

Our research model suggests that the quality of a redesigned model in a process redesign task will be a function of person-related characteristics (style and capacity) and process modeling competence (domain-relevant skills like process modeling knowledge and process modeling experience). According to [8, p. 1] we use two basic indicators of creativity: *fluency* is defined as "the ability to produce quantities of ideas which are relevant to the task instruction" and *originality* as "the ability to produce uncommon ideas or ideas that are totally new or unique". 'Fluency' is also often referred to as 'productivity' in creativity literature. In the context of business "originality is not enough. To be creative, an idea must also be appropriate - useful and actionable" [2, p. 78]. That's why we additionally use the variable *innovativeness*, which is defined as "the quality of an idea which is new and different" [27]. Fluency/productivity of ideas can be characterized in terms of amount and diversity of semantic categories to which ideas belong. The diversity of semantic categories is described as *breath* and the number of ideas per category is described as *depth* of creative production [19]. Unless noted otherwise, we expect similar effects for all five indicators for creativity of the process redesign.

Concerning the influence of creative style on creative redesign of a process we anticipate the following effects: Innovators are more likely to "break existing patterns" [16] and might produce a higher amount of innovative ideas. In contrast to innovators, adaptors perceive "boundaries less elastic and permeable" [12]. Therefore, we expect adaptors to make only small, continuous improvements to the model, leading to lower fluency, originality and innovativeness of their solutions. On the contrary, we expect innovators to radically change the process and do a complete reengineering. Thus, we conjecture that higher innovative style has a positive effect on the breath of creative production. Innovators might throw up a variety of ideas without going into detail, while adaptors might work out an idea in a very detailed way, and achieve a higher depth of creative production. In terms of our research model we expect:

- **Hypothesis 1:** Higher innovative style is positively associated with the fluency, originality, innovativeness and breath of creative production in a process redesign task.
- **Hypothesis 2:** Higher adaptive style is positively associated with the depth of creative production in a process redesign task.

Second, we turn to creative capacity. We hypothesize individual creative capacity to have a positive effect on all indicators of creative redesign. We state:

- **Hypothesis 3:** Creative capacity is positively associated with the creativity of the redesign.

Next, we consider process modeling competence. We speculate experience and knowledge in process modeling to contribute to the creative quality of process redesigns in terms of all indicators. Lower modeling experience might hinder creative persons from fully unfolding their capacity to provide creative ideas. Prior research has demonstrated that experience and education in the task domain are relevant for creative performance [13].

- **Hypothesis 4:** Process modeling knowledge is positively associated with creativity of the redesign.
- **Hypothesis 5:** Process modeling experience is positively associated with creativity of the redesign.

4 Empirical Study

To answer our research questions we conducted an empirical study using the Cheetah Experimental Platform [21]. Cheetah guides participants through a variety of questionnaire parts, offers a tutorial on process modeling and a process modeling tool for the redesign tasks, which logs every modeling action (e.g., adding and deleting of activities) to enable later analysis. The next paragraphs introduce main parts of the experimental design.

4.1 Measurement of Independent Variables

Creative Style: Innovative vs. Adaptive Problem-Solving Style (KAI (Kirton Adaption-Innovation Inventory): The Kirton Adaption-Innovation Inventory [11] measures individual problem-solving style relating to the quality of problem solutions. Respondents have to rate themselves on 32 items. It measures three different scales: sufficiency of originality, efficiency and rule governance.

Creative Capacity: Abbreviated Torrance Test of Creative Thinking (ATTA): The Torrance Tests of Creative Thinking (TTCT) measures divergent thinking and assesses the quantity and quality of creative ideas. It is a widely used (by over 2000 studies) instrument [8]. For the purpose of our study we use the verbal subscale of the abbreviated test version as a screening indicator of creative thinking abilities. The scores in the test are based on fluency (i.e., number of ideas) and originality (i.e., unusualness of ideas).

Process Modeling Competence: To measure process model competence participants are asked whether they have had any work experience with process models in practice and how many process models they have created or read. Additionally, we use a test on theoretical knowledge of process modeling developed by [15].

4.2 Measurement of Dependent Variables: Process Redesign

Experimental Redesign Tasks: In our experiment we ask participants to work out an improved TO-BE model for a given AS-IS process model of a coffee shop service. The AS-IS model was selected from a domain that it is understandable without any special domain knowledge. Participants got the process model (cf. Fig. 2) and the following instruction: “The coffee shop service wants to improve customer satisfaction by enhancing customer experience. How can the process be changed to implement that improvement?”. The task can be characterized as a measure-invoked redesign task targeting the measure- (customer) quality [26].

Measurement of Creativity of Process-Redesigns: For analyzing the creativity of the process redesigns (the TO-BE models developed by the participants) we selected relevant criteria from literature and developed a coding schema. Multiple indicators of creativity (i.e., fluency, originality, innovativeness, depth and breath of creative production) were captured as already described in the research model. The creativity of the redesign ideas was assessed by two independent experts according to selected creativity indicators as described below. We deployed an iterative consensus-building process to ensure validity and reliability of our assessment. All textual content produced by participants (labels of activities and conditions) was content-coded in semantic categories. Our first analysis was based on task and condition labels; future analysis will also include further process changes as rearranged tasks.

- **Fluency:** Number of ideas generated
- **Originality:** Based on the content coding we developed a scoring schema to assign points to ideas of participants. For instance we assigned 0.5 points for a refinement or a minor change of an existing task (e.g. "waiting time > 10 min" instead of "waiting time > 60 min" or "pay" instead of "pay cash"). We assigned 1 point to new labels if they were variations of existing tasks (e.g. "pay with credit card" is a variation of "pay cash"; "ask waitress for spices" is a variation of "get spices for coffee from add-on counter") or to activities which were very closely related to existing ones (e.g. "search for table" before "choose table"). In all variations either a verb or a noun in the new activity label was the same as in an existing activity label. For completely new ideas participants received 2 points each.
- **Innovativeness:** For ideas which were useful, purposeful and innovative for the process context and the specific task we assigned 1-2 points based on the raters' assessment. To ideas which were innovative but common (e.g. "read newspaper") we assigned 1 point, whereas to unique innovative ideas 2 points were assigned (e.g. "watching coffeemaker performing special tricks with coffee beans", "leave your coat in the cloakroom"). Ideas which cannot be influenced by the coffee shop management (e.g. "call a friend") were deemed as inappropriate in the context of the instruction and thus received no innovation points.
- **Breadth of creative production:** Number of semantic categories according to content categorization
- **Depth of creative production:** Average number of ideas per semantic category

Example of Measurement of Creativity of Process-Redesigns: In the following the creativity coding is illustrated along a redesign example of a study participant, as can be seen in Figure 2.

In the redesign solution we can see that the participant has added/changed 11 tasks or conditions resulting into 11 fluency points.

Concerning originality, "waiting time > 15 min" received only 0.5 points, as this is only a minor variation of the original label, which was commonly changed by participants (i.e., 37 out of 48 participants had changed that label). The activity "pay with EC-card" is a variation of "pay cash" and "ask for spices, if needed" is a variation of "get spices for coffee from add-on counter"; therefore, for both activities we assigned 1 point, respectively. All other new activities were scored with 2 points each.

The activities "receive cookie/coupon ... as a compensation", "read provided newspapers" and "watch TV (football games, music videos,...) on the big flat screen" were assigned 1 innovation point and "pay with special cafe bonus card (works with quick loading)" was assigned 2 bonus points, respectively. However, "talk to other guests" received no innovation point, since this was outside the sphere of influence of the coffee shop management.

Additionally, semantic categories were coded to determine the breath and depth of creative production. However, in contrast to other work on creativity measurement [8], we observed that in our data set the mere amount of nominations per semantic category was not an appropriate indicator for determining the creativity of an idea. For instance, activities "tip" and "eat" were only inserted once, but are not specifically creative. On the other hand, activities like "get coffee for free as a compensation for waiting time" (7 nominations) or "order online" (4 nominations) were new and innovative, but a higher amount of participants has had the same idea. A possible explanation for this may be that a higher sample size would be needed to determine uniqueness of ideas solely based on the amount of their nomination. Therefore, experts could only use the amount of nominations as an additional hint to score an idea.

4.3 Sample

A total of 50 business students participated in this study. Two students had to be excluded from the sample, because they had not worked on the process redesign task (they did not change anything in the models). Therefore, the final sample consisted of 48 participants. One student had not filled out the sample description items in the questionnaire, thus, we describe the remaining 47. Of these 47 respondents, 27 were males, 20 females. 92% were currently enrolled in a Bachelor program of business administration at a large European university. 65 % have already done any process modeling (e.g., with EPCs, BPMN, Flowcharts, Petri Nets) before taking part in the study.

5 Preliminary Results

In this section, we report the preliminary results of testing our sets of hypotheses. Hypotheses were tested using non-parametric correlations (Spearman). Table 1 shows the main results. We describe the results for each hypothesis in turn. For hypotheses 1 and 2, we first checked the reliability of the Kirton Adaption-Innovation Inventory scales for our sample. Internal consistencies of the scales were adequate ($\alpha_{\text{KAI originality}}$=0.63, $\alpha_{\text{KAI total efficiency}}$=0.81). ,$\alpha_{\text{KAI rule conformity}}$=0.71). Different to our expectations our data does not provide any support for hypothesis 1 and 2; none of the KAI subscales, nor the total score shows a significant positive correlation with any of the indicators for creativity of the redesign. Likewise, we obtained no support for hypothesis 3, which had predicted that creative capacity is positively associated with the creativity of the redesign solutions. The process modeling knowledge test score positively correlates as expected with originality and innovativeness, but not with the other indicators of creativity. This partially supports hypothesis 4. Hypothesis 5 predicted that experience with process models and creativity of the redesign solution are positively associated. Work experience with process models positively correlates with originality and innovativeness. The amount of process models read or created correlates positively with all creativity measures. Most strongly, the amount of process models read or created correlates with originality and innovativeness ($\rho = .342$ and $\rho = .398$). These results lend support to hypothesis 5. In summary, we obtained strong support for hypothesis 5, tentative support for hypothesis 4 and no support for hypothesis 1 - hypothesis 3.

Table 1. Correlations (Spearman's Rho, n=47-48), $^{+}$ sign. at $p < 0.1$, * sign. at $p < 0.05$

	Fluency	Originality	Innovativeness	Breadth of creative production	Depth of creative production
Creative Style: KAI total	-.021	.037	.100	-.068	.150
Creative Style: KAI originality	-.133	-.106	-.136	-.146	-.059
Creative Style: KAI total efficiency	.014	-.031	-.020	.033	-.056
Creative Style: KAI rule conformity	.032	-.009	-.137	.064	-.129
Creative capacity: ATTA total	.051	-.034	-.086	.048	.051
Creative capacity: ATTA fluency	.072	-.004	-.098	.084	.046
Creative capacity: ATTA originality	.004	-.075	-.049	-.018	.048
Process modeling knowledge test	.170	.264^{+}	.292*	.138	.173
Work experience with process models	.175	.251^{+}	.304*	.183	.149
Amount of process models read or created	.285^{+}	.342*	.398*	.257^{+}	.323*

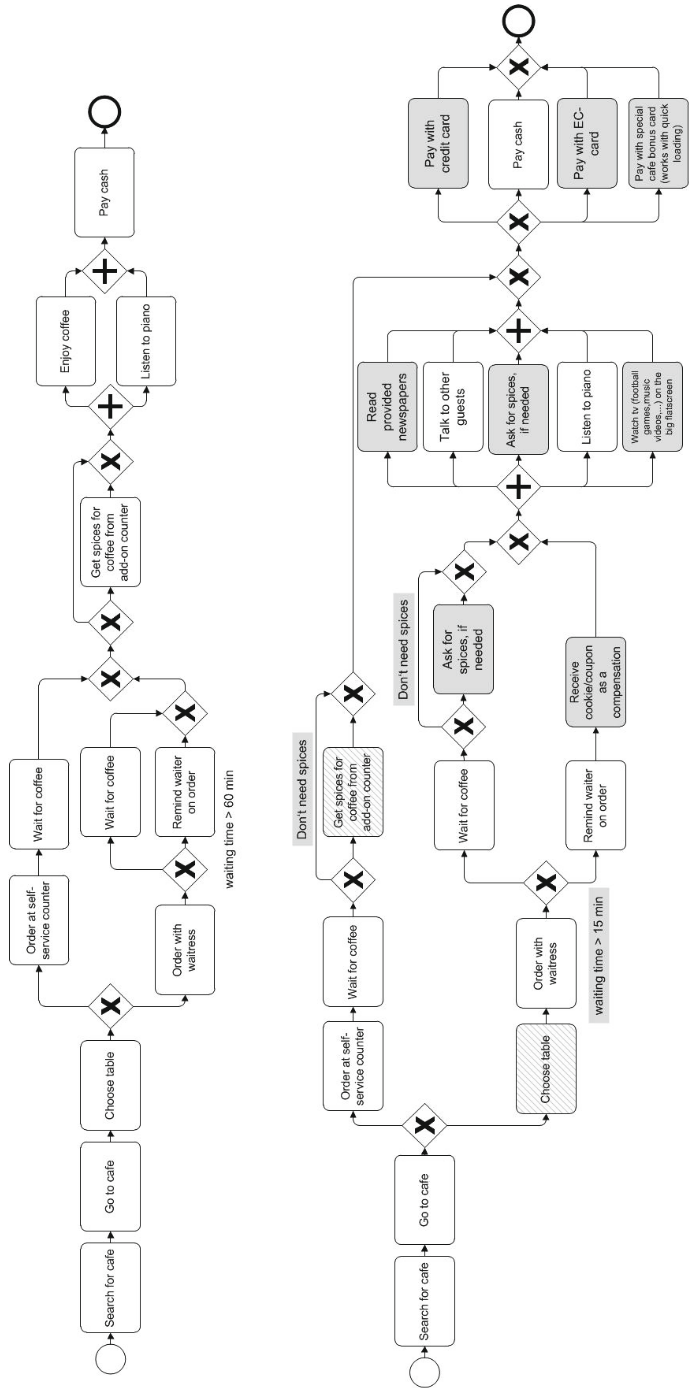


Fig. 2. Original Coffee Shop Model and Exemplary Redesign (*Grey: new or changed tasks and conditions, Shaded: change of task order*)

6 Discussion

This study provides preliminary empirical results on the associations between individual creative style and capacity and process modeling competence and creativity in a process improvement task. According to the results obtained, process modeling experience and knowledge seems to be a necessary precondition to enable people to redesign process models in a creative way. The quantitative measure 'amount of process models read or created', which was the most exact measurement of prior experience, was the best predictor for creativity of the redesigns.

Interestingly, we obtained strongest correlations with originality and innovativeness. These two indicators of creativity were not only simple automated measures quantifying creativity, but human expert rating was involved to capture the quality of the creative ideas. While it is easy to perform an arbitrary change to a process model, higher creative effort is necessary to really improve a process and a human rating seems to be necessary to determine if an idea is creative and useful for a particular context. Our data suggests that having some experience in process modeling is an important prerequisite to be able to produce a qualitatively good redesign of a process.

The results on creative style and capacity were counter intuitive, indicating that these individual characteristics did not have any correlation with the creativity of the process redesigns. A possible interpretation is that being familiar with the task setting - process modeling - is a pre-requisite to be able to bring in own ideas. As study participants were business students without extensive process modeling background, the newness of the task might have hindered students with low process modeling experience to materialize their creative abilities, so that presumably only participants with higher prior experience could produce original and innovative ideas. Future research is needed to examine how these associations change for process model experts in practice.

Next, we want to discuss several limitations of the study. First, the usual caveats associated with laboratory experiments limit the generalizability of the results.

Second, the large number of obvious and 'non-creative' additions and changes in the process models might indicate that some participants did not make a big effort to make their solutions creative in the redesign task. Maybe the instruction restricted creativity to some extend as only purposeful solutions were sought. Prior research can account for that explanation since explicit instruction to be creative improves the creative output of individuals [20]. However, we still think the instruction was good in the way we posited it since we were interested in the task setting of redesigning a business process in a realistic way.

Third, we want to discuss the fact that we had to exclude two participants from the data set as they did not change the business process at all. We can only speculate why this was the case. Either they did not understand what to do, because the instruction was open-ended and the large choice of possible changes was overwhelming for them or they were not motivated to participate in the study or they felt intimidated from using the modeling tool.

7 Conclusion and Future Work

We believe that our work serves as a valuable initial contribution for determining individual influence factors for creativity in process redesign tasks. In summary, our results indicate that process modeling competence is more relevant than individual creativity measured by existing creativity inventories. In future work we intend to validate our findings with further creative redesign tasks. Additionally, we would like to extend our analyses to include process change metrics as the amount of control flow changes, deletion and addition of routing symbols and edges of processes.

References

1. Amabile, T.M.: The social psychology of creativity: A componential conceptualization. Journal of Personality and Social Psychology 45(2), 357–376 (1983)
2. Amabile, T.M.: How to kill creativity. Harvard Business Review 76(5), 76–87 (1998)
3. Burroughs, J.E., Dahl, D., Moreau, C.P., Chattopadhyay, A., Gorn, G.J.: Facilitating and rewarding creativity during new product development. Journal of Marketing 75(4), 53–67 (2011)
4. Daniel Couger, J., Higgins, L.F., McIntyre, S.C.: (un)structured creativity in information systems organizations. MIS Quarterly 17, 375–397 (1993)
5. Davenport, T.H.: Process innovation: reengineering work through information technology. Harvard Business School Press, Boston (1993)
6. Ford, M.C.: A theory of individual creative action in multiple social domains. The Academy of Management Review 21(4), 1112–1142 (1996)
7. Foxall, G.R., Hackett, P.M.W.: The factor structure and construct validity of the kirton adaption-innovation inventory. Personality and Individual Differences 13(9), 967–975 (1992)
8. Goff, K., Paul Torrance, E.: Abbreviated Torrance Test for Adults (ATTA). Manual. Scholastic Testing Service, Bensenville, Illinois (2002)
9. Gartner Group. Leading in times of transition: The 2010 cio agenda. Technical report, Gartner, Inc. (2010)
10. Hammer, M., Champy, J.A.: Reengineering the Corporation: A Manifesto for Business Revolution. Harper Business Books (1993)
11. Kirton, M.J.: Adaptors and innovators: A description and measure. Journal of Applied Psychology 61(5), 622–629 (1976)
12. Kirton, M.J.: Have adaptors and innovators equal levels of creativity? Psychological Reports 42, 695–698 (1978)
13. Martinsen, O.: Insight problems revisited: The influence of cognitive styles and experience on creative problem solving. Creativity Research Journal 6(4), 435–447 (1993)
14. McAdam, R., McClelland, J.: Individual and team-based idea generation within innovation management: organisational and research agendas. European Journal of Innovation Management 5, 86–97 (2002)
15. Mendling, J., Strembeck, M.: Influence factors of understanding business process models. In: Proc. BIS 2008, pp. 142–153. Springer (2008)
16. Mudd, S.: Kirton adaption-innovation theory: organizational implications. Technovation 15(3), 165–175 (1995)

17. Müller-Wienbergen, F., Müller, O., Seidel, S., Becker, J.: Leaving the beaten tracks in creative work - a design theory for systems that support convergent and divergent thinking. Journal of the Association for Information Systems 12(11) (2011)
18. Mumford, M.D., Simonton, D.K.: Creativity in the workplace: People, problems, and structures. The Journal of Creative Behavior 31(1), 1–6 (1997)
19. Nijstad, B.A., Stroebe, W., Lodewijkx, H.F.M.: Cognitive stimulation and interference in groups: Exposure effects in an idea generation task. Journal of Experimental Social Psychology 38(6), 535–544 (2002)
20. O'Hara, L.A., Sternberg, R.J.: It doesn't hurt to ask: Effects of instructions to be creative, practical, or analytical on essay-writing performance and their interaction with students' thinking styles. Creativity Research Journal 13(2), 197–210 (2001)
21. 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)
22. Reijers, H.A., Liman Mansar, S.: Best practices in business process redesign: an overview and qualitative evaluation of successful redesign heuristics. Omega 33(4), 283–306 (2005)
23. Seidel, S., Müller-Wienbergen, F., Becker, J.: The concept of creativity in the information systems discipline: Past, present, and prospects. Communications of the Association for Information Systems 27(1), 217–242 (2010)
24. Seidel, S., Müller-Wienbergen, F., Rosemann, M.: Pockets of creativity in business processes. Communications of the Association for Information Systems 27 (2010)
25. Seidel, S., Rosemann, M.: Creativity management - the new challenge for bpm. BPTrends (2008)
26. Shtub, A., Karni, R.: Business process improvement. In: ERP, pp. 217–254. Springer, US (2010)
27. Web Finance, Inc. Innovative (2012)
28. Woodman, R.W., Sawyer, J.E., Griffin, R.W.: Toward a theory of organizational creativity. Academy of Management Review 18(2), 293–321 (1993)

User-Centric Abstraction of Workflow Logic Applied to Software Engineering Processes

Gregor Grambow[1], Roy Oberhauser[1], and Manfred Reichert[2]

[1] Computer Science Dept., Aalen University
{gregor.grambow, roy.oberhauser}@htw-aalen.de
[2] Institute for Databases and Information Systems, Ulm University, Germany
manfred.reichert@uni-ulm.de

Abstract. Software development is a dynamic, complicated, and labor-intensive undertaking. Numerous software engineering process models have been created and applied to address its complexity, schedule pressure, and product quality. These process models are rather abstract and not directly operationally relevant for the software engineers executing these processes, since they mostly provide relatively coarse-grained work packages and lack fine-grained user-centric workflows directly supporting users. Such user-centric workflows have been difficult to implement in an automated fashion as they are very dynamic and user acceptance for both modeling and prescribing such fine-grained activities is fairly low. This paper provides an approach to abstractly model user decisions influencing the actual trace of such automated workflows. By hiding internal complexity, communication with users is simplified while supporting required flexibility. This contributes towards removing hindrances and enabling the application of and user acceptance for automated user-centric workflows in software engineering and in domains exhibiting similar issues.

Keywords: Human-centric Process-Aware Information Systems, User-centric Workflows, Process-Centered Software Engineering Environments.

1 Introduction

Software development is a complicated process, and software engineering (SE) projects continue to face challenges and struggle with inherent difficulties [1][2][3]. The high dynamicity inherent to that discipline and the intangibility of the developed product hamper consistent process management. Furthermore, the developer must address various requirements concerning intellectual efforts and dynamic collaboration with others. Much of this remains undetected, untraced, and ungoverned. However, in various domains it has been shown that process orientation and explicit process management can be beneficial [4][5]. Process management fosters project efficiency [6] and product quality [7].

Based on such knowledge, various process models for SE have been developed. Examples include the Unified Process [8] and its variants or the V-model XT [9]. They support establishing, documenting, and tracking the SE process. Yet several

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 307–321, 2012.

problems occur in process execution: Most SE process models remain too general by describing abstract work packages. On the operational level, where individuals interact with tools and among each other to actively create software, SE process models lack concrete intentional support [10]. Thus, it is problematic to govern the SE process using such a model and support the people in their everyday work, as the process does not really "touch" them. Thus, maintenance of the processes model incurs additional overhead for the participants, because alignment of the abstract process with operational reality has to be manually established and tracked. Furthermore, process execution is not actively integrated into project execution. The process is not automatically aligned to events happening in various project situations, while the integration of information between process management and other areas like quality or knowledge management remains difficult.

Our previous work has addressed a number of these challenges: Facilities were developed that enable the acquisition and integration of project context data with process execution [11]. Execution semantics for process management have been extended to enable the mapping of process models and the integration of different levels of user activities. The automated exchange of information between process execution and areas like knowledge management [12] and quality management [11] has also been established. However, SE processes remain too abstract, and a tool seeking to truly support software engineers holistically in their projects should be aware of what they are actually doing. Thus, user-centric workflows are required. This notion is explained in the following:

In our context, workflows denote concrete sequences of activities, while (SE) processes are rather abstract, concerning activity sequencing but also integrating other aspects (such as the team or organizational structure). User-centric workflows are workflows that describe or govern concrete activities of humans (like, in SE, creating a new software function). This implies, on the one hand, that all activities of such a workflow contain task information guiding users (for tasks like, in SE, creating a software build from some source code). On the other hand, as the workflow shall govern what the user concretely does, the latter needs to influence and control the actual trace of the workflow that is required to achieve the goal.

To enable reasonable automatic governance and support for user-centric workflows, the following requirements would have to be satisfied:

- The granularity of workflow activities in the workflow should be fine enough to make the governing system aware of what the user is really doing, yet coarse enough not to overburden the user with copious micro-activities.
- The user should have control over the decisions in the workflow as that workflow is used for governing his activities.
- Workflow interaction shall not distract the user from his or her actual work.

The remainder of this paper is structured as follows: Section 2 presents a concrete problem scenario, Section 3 gives insights on the developed approach, and Section 4 shows its practical application to the problem scenario. The paper concludes in Section 6 after discussing related work in Section 5.

2 Problem Scenario

This section demonstrates shortcomings of contemporary process management systems that can cause problems when trying to model user-centric workflows. As SE is a dynamic discipline, it epitomizes such a workflow. For this paper, we create a small example using experiences we gained while working with software development companies. The example concerns a workflow that governs activities utilized for creating a new piece of software. The workflow shown in Fig. 1 is a simplified version of a workflow we created during an interview with software developers at a company that produces software.

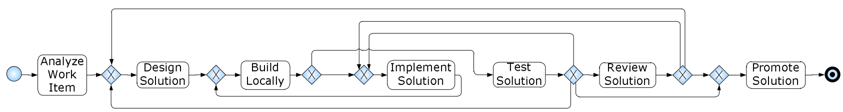


Fig. 1. Example workflow (unstructured)

The workflow starts with the analysis of the work item (assignment) to determine what has to be done to achieve the work item goal (e.g., create the desired functionality) followed by the design of the solution. Thereafter, the code of the system is checked out and a local build is created. Additionally included activities concern the implementation as well as testing and a review of the solution. The workflow concludes with the promotion of the developed solution, meaning the integration of the locally created source code into the mainline development branch within the shared version control system (VCS). As the described activities in most situations cannot be simply executed in a basic linear non-repeating sequence, the workflow contains several loops.

Note that the workflow is not well structured according to [13], since loops overlap in some cases, contradicting proper nesting of workflow elements. Unstructured workflows have been proven to be badly readable and error prone. Thus, the workflow was restructured to enable proper nesting of the elements as shown in Fig. 2. This has been done by duplicating the build activity to resolve the overlapping loops. This configuration was chosen based on domain knowledge, since a real difference between the two new building activities exists: The first one, *Initial Local Build*, is conducted once before any implementation changes are applied to verify that the checked out versions of the code files build on the local machine (a.k.a. sandbox or programmer's directory). The second activity, *Build Locally*, is conducted to verify that the changed code is buildable.

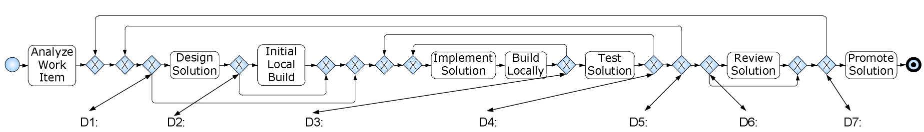


Fig. 2. Example workflow (structured)

The workflow shows the different decisions that have to be connected to certain variables to govern the sequencing of the activities. Each decision could be connected to a question towards the user to acquire necessary and missing information. Note that this realization of the workflow shows several flaws: while processing the workflow, the user always has to provide values for all decisions except D2 (this one can be set internally as the initial build activity should be only executed once). Consider activity *Test Solution:* it has four possible successors (*Review Solution*, *Promote Solution*, *Design Solution*, and *Implement Solution*), but the user has to be involved in all decisions in the workflow. This kind of workflow governance can be confusing and cumbersome for the user, provoking mistakes on user decision input that then result in workflow traces that are inconsistent with reality. This type of workflow interaction will most likely be perceived as burdensome by users, which may affect their overall impression and resulting acceptance or rejection of such an automated guidance system. To resolve this, a more user-centric way of communication is desirable, letting the user focus on what she or he is doing rather then being distracted by the supplementary information requirements of a workflow that should be assisting them.

3 Abstract User Decision Mapping

This section elucidates the solution for the aforementioned problem starting with a brief introduction on the on the framework that forms the basis for that solution.

3.1 Solution Basis

To address the aforementioned challenges, we have developed a framework called CoSEEEK (Context-aware Software Engineering Environment Event-driven frameworK) [14]. With holistic SE process support as a primary goal, the framework was designed to be an active component in that process supporting vertical and horizontal integration of activities in a project. In this context, vertical means abstract to concrete levels of activities while horizontal means integration of different project areas like, e.g., quality management and software development. The framework integrates various technologies to create an infrastructure that can extract, unify, and utilize context, process, and project information in an automated fashion. In the following, different components of the framework are briefly explained to introduce the background for the solution approach presented in this paper:

To be able to provide automatic workflow governance, the *Process Management* component integrates a Process-Aware Information System (PAIS) that enables workflow execution and enforces correctness criteria in this area. That way the basis for mapping parts of the SE process models to automatically governed and supported workflows is set. As SE workflows are dynamic, a dynamic PAIS (Aristaflow BPM suite [15]) has been integrated to enable the adaptation of running workflow instances.

The *Context Management* component constitutes the core of the framework. It not only enables information storage, but also active usage of that information for automation. Integrated semantic technology (ontology and reasoning capabilities) enables automatic information processing. In this component, extensions to the process management concepts are stored that enable the mapping of SE process models with all of their diverse additional information and dependencies. Via these extensions, the *Context Management* component has direct access to process execution and can use additional context information to optimally align the process with the actual project context.

The *Event Management* component collects and processes environmental information. Event sensors integrated into various SE tools (like Integrated Development Environments or VCS) generate events for the *Event Management* component. That component, via complex event processing, combines and distributes events to the *Context Management* component for contextual assimilation. The *Quality Management* component, in turn, enables an active connection between software quality management and process management. That way, it is possible to automatically detect problems, determine proper software quality measures for them, and integrate them into running workflows. Finally, with the *Knowledge Management* component, there is also an active connection between project knowledge and executed process. Users can collect knowledge in a semantic Wiki and the system automatically injects that information into process execution where appropriate.

We have briefly explained how process models are mapped, how the process is integrated with real time information from the SE projects, and how the process is unified with other project areas like quality or knowledge in an automated fashion. However, to truly connect process execution with users in SE, user-centric workflows are required that guide users and associate their activities with the abstract process. Yet to overcome the obstacles described in Section 2, connections between the *Context Management* and *Process Management* components are utilized as illustrated in Fig. 3. All basic process management concepts are mirrored in the *Context Management* component and are tightly connected by the framework. These concepts enable the annotation of process management with contextual information, while enabling the framework to actively influence process management.

3.2 User Decision Modeling Concepts

Fig. 3 illustrates the mappings of the process management concepts existing in the *Context Management* component using a simple example: A *Work Unit Container Template* mirrors a workflow template and a *Work Unit Template* mirrors an activity of a workflow template. That way, the *Context Management* component is aware of the workflow templates and can automatically instantiate new workflows from them. The latter are also mirrored using the *Work Unit Container* and the *Work Unit*. These concepts give the *Context Management* component enhanced control over workflow execution and thus enable it to completely encapsulate and extend the *Process Management* component and to provide high level context-aware workflow governance to the users (see [12, 13] for further details).

The approach taken to enable abstraction from internal process logic and variables is grounded on these concepts. The *Work Unit Container Template* is extended by

Workflow Variable Templates that map the workflow variables defined in the workflow template. For these variable templates, the *Workflow Variable Values* provide possible pre-defined values. The latter can then be used to set the variables for a workflow instance during execution. These are mapped by the *Workflow Variables*. Each *Work Unit Container Template* has a set of *Workflow Variable Templates* that provide the initial values for the variables of a new workflow instance. To grant the user flexible and abstract control about the workflow variables, the *Workflow User Information* and *User Decision Alternative* are used: The former is connected to a *Work Unit Template* and used to inform the user executing that activity about a decision the user has to make. The latter represents one alternative of such a decision, and is, in turn, connected to *Workflow Variable Values* that are used when that alternative is chosen by the user.

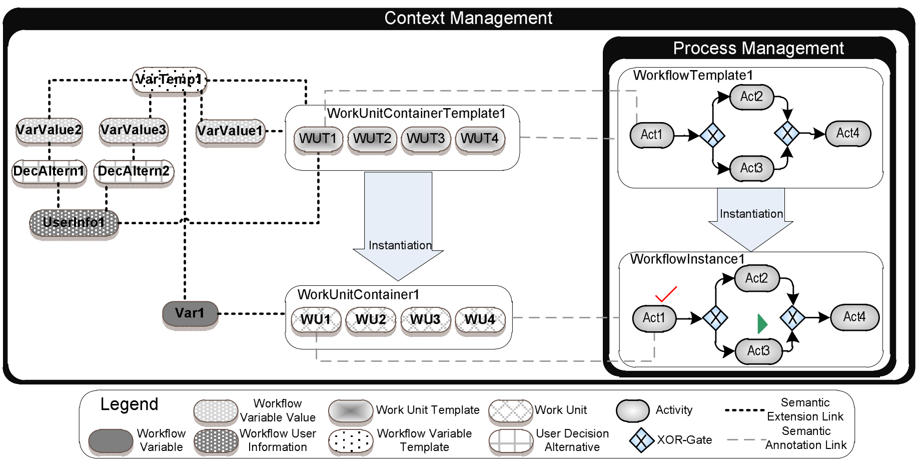


Fig. 3. Extensions to workflow governance

Consider the simple example in Fig. 3: WorkflowInstance1 contains four activities from which two (*Act2* and *Act3*) are mutually exclusive. For example, they could be two different code review activities (e.g., a peer review and a code inspection, both having different properties regarding required effort, duration, and error detection rate). As the user has to decide which one is appropriate in the current situation, a value for the variable that is used for the XOR-gate in the workflow has to be acquired from the user. In this example, *UserInfo1* that is connected to the *Work Unit Template* that maps the first activity in the workflow would gather information on the upcoming decision while the user executes the current activity (e.g., asking "How much time is available for reviewing your code?"). The two options for that decision are modeled by *DecAltern1* and *DecAltern2* that provide the values for the XOR-gate using *VarValue2* and *VarValue3* (e.g., Sufficient time left / High schedule pressure). This simple example is used for illustration; to address the problem scenario of Section 2, more complex mappings are shown in the next section.

To have a sustainable basis for the approach, the operational semantics of the described concepts rely on the definitions in Table 1. Due to space limitations, not all definitions are shown in this paper. The basic mapping and extension of concepts within a PAIS has been described in previous work [11].

Table 1. Definitions

Concept	Description
Identifiers	All valid identifiers over a given alphabet. All concepts have a name ∈ *Identifiers*
Types	All definable object types. All concepts have a distinct type ∈ *Types* that is defined by the tuples in the following definitions.
WFTemplates	All workflow templates within a PAIS
ActivityTemplates	All activities within workflow templates in a PAIS
WFInstances	All workflow instances within a PAIS
ActivityInstances	All activities within workflow instances in a PAIS
WorkUnitContTempls	All *work unit container templates* that are used to map and extend the workflow templates of a PAIS
WorkUnitTempls	All *work unit templates* that are used to map and extend the activity templates of the integrated PAIS
WorkUnitConts	All *work unit containers* that are used to map and extend the workflow instances of the integrated PAIS
WorkUnitTempls	All *work units* that are used to map and extend the activities of a PAIS

On this basis, the following definitions are made for the concepts used for the mapping of internal workflow variables to high-level user decisions. These formal definitions support the exact description of each concept and its relations, and enable the further definition of conditions that guarantee correct executability. The definitions assume that each concept has a type ∈ *Types* and a name ∈ *Identifiers*.

Definition 1. A ***workflow user information*** represents one decision that a user can make when processing an activity of a workflow. It is a tuple workflowUserInfo = (type, name, decAlternSet, workUnitTempl, info) where

- decAlternSet is a finite set of decision alternatives with decAltern ∈ *DecAlternatives* (cf. Def. 2).
- workUnitTempl ∈ *WorkUnitTempls* is the work unit template to which workflowUserInfo belongs.
- info ∈ STRING is the information about the decision for the user.

WorkUserInfos describes the set of all definable workflow user information.

Definition 2. A ***user decision alternative*** represents one alternative for a decision decision a user can make when processing an activity of a workflow. It is a tuple decAlternative = (type, name, userInfo, varValueSet, decInfo, decId, standard) where

- userInfo ∈ *WorkUserInfos* is the workflow user information to which decAlternative belongs.

- varValueSet is a finite set of workflow variable values with varValue ∈ *VarValues*.
- decInfo ∈ STRING is the information for the decision alternative.
- decId ∈ INTEGER is the id for the decision alternative.
- standard ∈ BOOLEAN marks the initially selected decision alternative.

DecAlternatives describes the set of all definable user decision alternatives.

Definition 3. A ***workflow variable value*** represents a value to which a workflow variable can be set in a certain situation. It is a tuple varValue = (type, name, varTempl, workUnitContTemppl, decAlternative, value) where

- varTempl ∈ *VarTempls* is the workflow variable template whose corresponding workflow variable is set by varValue.
- workUnitContTempl ∈ *WorkUnitContTempls* ∪ {NULL} is the work unit container template for whose instances varValue provides an initial value for a variable; will be NULL in case varValue is supplied for a decision alternative
- decAlternative ∈ *DecAlternatives* ∪ {NULL} is the decision alternative for which varValue provides a value for a workflow variable; will be null in case varValue is supplied for a work unit container template
- value ∈ INTEGER is the value to be used for the variable.

VarValues describes the set of all definable workflow variable values.

Definition 4. A ***workflow variable template*** represents the definition of the mapping of a workflow variable that exists for a workflow template within a PAIS. It is a tuple varTempl = (type, name, workUnitContTempl, paisVarName) where

- workUnitContTempl ∈ *WorkUnitContTempls* is the work unit container for which varTempl represents a variable.
- paisVarName ∈ STRING is the value to be used for the variable.

VarTempls describes the set of all definable workflow variable templates.

Definition 5. A ***workflow variable*** represents the mapping of a workflow variable that exists for a certain workflow instance within a PAIS. It is a tuple workVar = (type, name, workUnitCont, varTempl, currentValue) where

- workUnitCont ∈ *WorkUnitConts* is the work unit container for which varTempl represents a variable.
- varTempl ∈ *VarTempls* is the variable template to which workVar belongs.
- currentValue ∈ INTEGER is the current value of the variable.

WorkVars describes the set of all definable workflow variables.

These five definitions provide the basis for the concept developed in this paper. To ensure executability, some basic conditions have to be satisfied. These have been defined in the following: for all variables in a workflow, mappings must exist (cf. Def. 6 b), their association to *Workflow Variable Templates* must be clear (cf. Def. 6

a), and initial values should be provided (cf. Def. 6 c) so that all workflows can be executed without user communication. The latter should also be well defined, meaning that for each decision, one or more alternatives exist (cf. Def. 6 d) from which one is set as the default (cf. Def. 6 e), and all have variable values to set if a choice exists (cf. Def. 6 f), since a user decision with no impact would be irrelevant.

Definition 6. Let workUnitCont = (type, name, wfInstance, workUnitSet, assignment, mandInputSet, outputSet, roleSet, flowVarSet, basis) $\in$ WorkUnitConts be a *work unit container* belonging to a project within the system. Then:

a) $\forall$varTempl: varTempl.paisVarName = wfTemplate.variable.name, i.e., the names of the variables in the PAIS and the mapping with *variable templates* must match.
b) $\forall$WFtemplate.variable: $\exists$varTempl $\equiv$ WFtemplate.variable, i.e., there exists a mapping *workflow variable template* for all variables in a workflow template.
c) $\forall$varTempl $\in$ workUnitContTempl: $\exists$varValue with varValue.varTempl = varTempl, i.e., all *variable templates* within a *work unit container template* shall have an initial value.
d) $\forall$workflowUserInfo: |workflowUserInfo.decAlternSet| $\geq$ 1. i.e., every *workflow user information* must have at least one *decision alternative.*
e) $\forall$workflowUserInfo: |decAlternative $\in$ workflowUserInfo.decAlternSet with decAlternative.standard = TRUE| = 1, i.e., for each *workflow user information* there has to be exactly one standard alternative.
f) $\forall$decAlternative with |workflowUserInfo.decAlternSet| > 1:
|decAlternative.varValueSet| $\geq$ 1, i.e., if a *workflow user information* has more than one *decision alternative*, each *decision alternative* must set at least one variable.

Utilizing these definitions, a workflow instance is always provided with the necessary values of all the variables used to govern its execution. When such an instance is created from a workflow template, all variables receive their initial values from the *workflow variable values* defined for the *work unit container*. These values shall be defined in a way to represent the typical trace of the workflow, minimizing the necessary user interaction required for appropriately governing the instance. That user interaction is illustrated in Fig. 4. As aforementioned, the *Context Management* component adds several types of information to the workflows of the *Process Management* component. In this case, information about the users' activities is relevant. It comprises an *Assignment* that represents information about the activity for which the workflow was initiated, e.g., 'Develop new feature X'. The different activities to reach that goal are represented by *Assignment Activities*, as, e.g., 'Implement Solution'. To enable better operation assistance, the concept of the *Atomic Task* was also added, representing activities like checking in source code or running unit tests. For more information on these concepts, we refer to [11].

The different steps taken by the system to extend workflow execution with more user-related semantics are now described, and illustrated in Fig. 4:

Step 1. An event (by a user or the system) causes a workflow to start.
Step 2. The workflow execution reaches one or more activities that become enabled. This information is distributed to the *Context Management* component.
Step 3. The *Context Management* component, in turn, distributes the relating *Assignment Activity* and potential additional information to the user.
Step 4. The user starts the processing of the *Assignment Activity.*
Step 5. The *Context Management* component retrieves the decision information and its alternatives and distributes it to the user.
Step 6. The user selects one decision alternative (if different from the pre-selected).
Step 7. The user finishes the processing of the *Assignment Activity.*
Step 8. The *Context Management* component informs the *Process Management* component that the active activity may complete now. This information incorporates values for the workflow instances variables.
Step 9. The Process Management component sets the values of the variables and then lets the activity complete and the workflow instance continue.

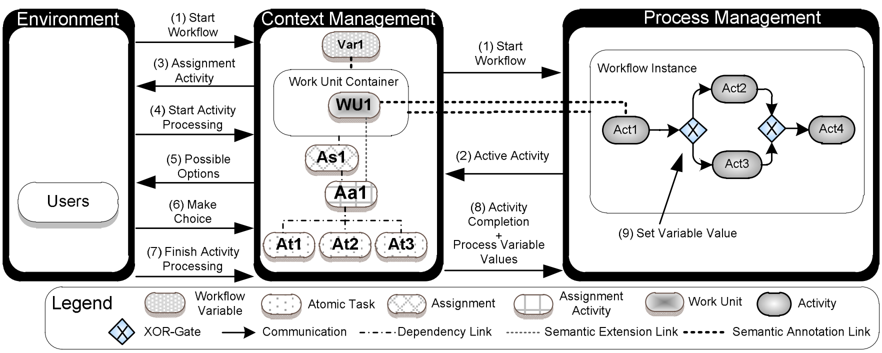


Fig. 4. Workflow enactment with extensions

This concept offers flexibility for transferring workflow governance control to users by mapping internal workflow variables to user decisions. There can be multiple ways of defining and connecting different concepts to match projects needs:

- A simple 1-1 mapping of *User Decision Alternatives* to variables. With such a mapping, a user could directly set the value for each workflow variable. In this case the *Workflow User Information* would be used to store an appropriate question or statement to inform the user what he is about to control now. The *User Decision Alternatives*, in turn, would store textual information on *decision alternatives* such as 'Is additional implementation effort still required? – Yes/No' which is less error-prone than setting a variable like 'AdditionalImplEffort – true/false';

- A more complex n-m mapping where each *User Decision Alternative* sets multiple variables to provide the user with support for a more abstract decision. Each of the alternatives could even set different sets of variables, allowing completely different traces to be produced. This way of mapping could make big complicated workflows easier to handle for users having one consistent decision with a limited set of alternatives per activity instead of having to set multiple variables all the time;
- With the abstraction from the variables and the explicit modeling of user decisions, there is the possibility of restricting certain options of a decision that would be available if there was direct access to the workflow; and
- Since the user-decision modeling is user centric and abstracted from the workflow internals, it can also serve for hiding technical complexity inherent to the workflows. Well-structured workflows are often bigger than not well-structured workflows describing the same situation (as shown in Section 2 with an additional activity). They may be more comprehensible for the modeler, but could be more complicated for the executing person. This can be compensated in our approach via the additional abstraction layer introduced towards the user.

4 Application Scenario

This section shows the proposed approach applied to a concrete example in regard to the scenario of Section 2. In that example, five user decisions were required to properly govern the workflow (D2 can be set internally and D3 and D4 could be consolidated), all of which had to be shown to the user during the entire workflow execution. To simplify this, the following mapping of the internal workflow decisions was created: for each possible successor of an activity, a decision alternative was created. That way the user can directly choose which activity to do next while executing an activity. Thus, no additional information on the decision is necessary (and the info string of the workflow user information can remain empty). Each of the decision alternatives then sets exactly the set of workflow variables required to activate the chosen activity. Additionally, a set of initial workflow variables sets the workflow variables to the most likely trace to minimize the required user interactions. Fig. 5 shows the CoSEEEK user GUI associated with the Section 2 workflow governed and the relevant concepts connecting workflow governance with the GUI.

The upper section of the GUI is reserved for special context and guidance information, e.g., for coordination [16] or quality [11] support. The lower part of the GUI, which is the focus of this paper, shows the activity section. The latter provides the user with valuable information about the process in which the user assignment is contained. This section also provides the user with current and future activity information: The current activity is shown as well the options for the next activity. The user can then simply click on one of these to choose it as successor of the current one. That way, internal workflow variables are completely hidden from the user. Fig. 5 shows the GUI when the user processes the *Test Solution* activity: the user has four possible next choices and each of them affects a different set of variables. For example, by choosing Design Solution as the successor, only decisions D1, D4, and D5 need to be set with the correct variable values.

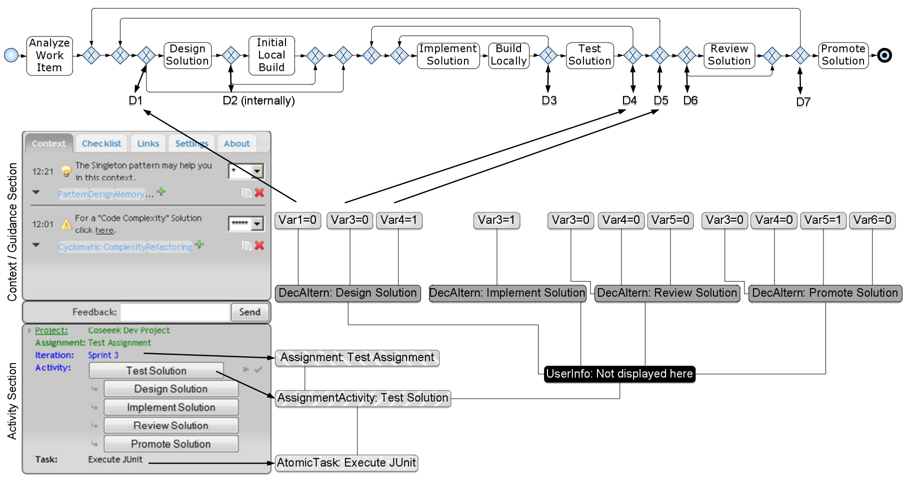


Fig. 5. Workflow GUI

5 Related Work

In related work, there have been attempts to enable automatic support for users in the software development process. This started in the 1990s with Computer-Aided Software Engineering (CASE). Approaches like [18] tried to combine various tools for different activities to create more holistic support, yet a process component was absent. This issue was addressed by Process-Centered Software Engineering Environments (PCSEEs) like [19], which not only supported single activities but also considered their relations and impact on the produced product. Recent developments include Application Lifecycle Management (ALM) tools that connect different areas of the projects via the produced product, notably IBM Jazz [20]. While these approaches aim for active and holistic automated support, none incorporates user-centric support for workflows as shown in this paper.

Concerning abstraction from process management internals, there has been a fair amount of related work: For example, [21][22] provides users with abstractions from technical process specifications using views. Similarly, [23] and [24] simplify process models to generate views for better comprehension. [23] abstracts parts of the model using semantic similarity of activities using structural information of the models. [24] applies a two-phase procedure for aggregating parts of process models. Although these approaches deal with the abstraction of process models, they only aid the users by providing better views of the models and not in abstracting and simplifying the communication with users. In contrast, the approach presented in this paper contributes a technique for transferring workflow control towards the users by making communication with the process more convenient and less error-prone.

The approach presented in [25] incorporates several extensions to process management concepts similar to this approach. However, the former deals with design-time process model configurations in contrast to the dynamic runtime process support CoSEEEK offers.

6 Conclusion

SE is a dynamic discipline that highly relies on the participating individuals, their intellectual work, and their collaborations. It thus poses a challenge towards automating process management that implements SE process models to an operational level. In our previous work, we implemented the basis for extending process management concepts to enable the enrichment of the processes. This paper adds a way of communicating between the process models and the users that is oriented towards user needs and preferences. By abstracting from process management internals, the communication can be modeled in a way that better assists users. The mapping from internal process variables to user decisions adds flexibility and simplifies the communication with the user. The application to a user-centric software development workflow yields the following benefits: Easy pre-configuration of the variables is possible. The amount of communication with the user can be reduced. Complex mappings of variables to decision alternatives become possible. Moreover, the system has improved options for traceability and support, since it is aware of the users' intent so that dynamic workflows can be governed without taking the users' freedom away.

Based on this support for abstract user decision modeling, future work will not only include currently ongoing industrial investigation, but also extend this concept to enable the system to exploit more of its available context data to improve the user experience. This includes better alignment of the workflow to the current situation (e.g., skill of the user); extending the initial set of variable values with multiple sets of initial values (note that since such properties cannot be known a priori, the dynamic alignment is applied just in time when the workflow is started); supporting additional influencing factors (e.g., the quality goals of a project) to enable the same user decision to produce different traces for different goals (like, e.g., choosing a more effective review activity if goals like reliability and maintainability are important); and the inclusion of other situational properties (schedule pressure, criticality, etc.) that have been modeled in [17] and will be also connected to the user decision modeling.

References

1. Brooks, F.P.: No Silver Bullet: Essence and Accidents of Software Engineering. Information Processing (1986)
2. Yourdon, E.: Death March, 2nd edn. Prentice Hall, Upper Saddle River (2004)

3. Jones, C.: Get Software Quality Right. Dr. Dobb's Journal, June 28 (2010)
4. Lenz, R., Reichert, M.: IT support for healthcare processes-premises, challenges, perspectives. Data & Knowledge Engineering 61(1), 39–58 (2007)
5. 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)
6. Gibson, D.L., Goldenson, D.R., Kost, K.: Performance results of CMMI-based process improvement. Technical Report, Software Engineering Institute, Carnegie-Mellon University, Pittsburgh (2006)
7. Heravizadeh, M.: Quality-aware business process management. PhD Thesis, Queensland University of Technology (2009)
8. Jacobson, I., Booch, G., Rumbaugh, J.: The unified software development process. Addison-Wesley (1999)
9. Rausch, A., Bartelt, C., Ternité, T., Kuhrmann, M.: The V-Modell XT Applied–Model-Driven and Document-Centric Development. In: Proc. 3rd World Congress for Software Quality, vol. III, pp. 131–138 (2005)
10. Wallmüller, E.: SPI-Software Process Improvement mit Cmmi und ISO 15504. Hanser Verlag (2007)
11. Grambow, G., Oberhauser, R., Reichert, M.: Contextual Injection of Quality Measures into Software Engineering Processes. Int'l Journal on Advances in Software 4(1 & 2), 76–99 (2011)
12. Grambow, G., Oberhauser, R., Reichert, M.: Towards Dynamic Knowledge Support in Software Engineering Processes. In: Proc. 6th Int'l Workshop on Applications of Semantic Technologies, AST 2011, held in conjunction with INFORMATIK 2011 (2011) (accepted for publication)
13. Mendling, J., Reijers, H.A., van der Aalst, W.M.P.: Seven process modeling guidelines (7pmg). Information and Software Technology 52(2), 127–136 (2010)
14. Oberhauser, R.: Leveraging Semantic Web Computing for Context-Aware Software Engineering Environments. Semantic Web, pp. 157–179. In-Tech, Vienna (2010)
15. Dadam, P., Reichert, M.: The ADEPT project: a decade of research and development for robust and flexible process support. Computer Science-Research and Development 23(2), 81–97 (2009)
16. Grambow, G., Oberhauser, R., Reichert, M.: Enabling Automatic Process-aware Collaboration Support in Software Engineering Projects. In: Selected Papers of the ICSOFT 2011 Conference. CCIS (2011) (accepted for publication, 2012)
17. Grambow, G., Oberhauser, R., Reichert, M.: Contextual Generation of Declarative Workflows and their Application to Software Engineering Processes. Int'l Journal on Advances in Intelligent Systems 4(3 & 4) (accepted for publication, 2012)
18. Emmerich, W., Kroha, P., Schäfer, W.: Object-Oriented Database Management Systems for Construction of CASE Environments. In: Wagner, R.R., Lazanský, J., Mařík, V. (eds.) DEXA 1993. LNCS, vol. 720, pp. 631–642. Springer, Heidelberg (1993)
19. Conradi, R., Hagaseth, M., Larsen, J.O., Nguyen, M., Munch, B., Westby, P., Zhu, W., Jaccheri, M., Liu, C.: Object-oriented and cooperative process modelling in EPOS. In: Software Process Modelling and Technology, pp. 9–32. Research Studies Press Ltd., Taunton (1994)
20. IBM Jazz (received January 2012), `https://jazz.net/`
21. Bobrik, R., Reichert, M., Bauer, T.: View-Based Process Visualization. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 88–95. Springer, Heidelberg (2007)

22. Reichert, M., Kolb, J., Bobrik, R., Bauer, T.: Enabling Personalized Visualization of Large Business Processes through Parameterizable Views. In: Proc. 27th ACM Symposium on Applied Computing (2012) (accepted for publication, 2012)
23. Smirnov, S., Reijers, H.A., Weske, M.: A Semantic Approach for Business Process Model Abstraction. In: Mouratidis, H., Rolland, C. (eds.) CAiSE 2011. LNCS, vol. 6741, pp. 497–511. Springer, Heidelberg (2011)
24. Eshuis, R., Grefen, P.: Constructing customized process views. Data & Knowledge Engineering 64(2), 419–438 (2008)
25. La Rosa, M., Dumas, M., Ter Hofstede, A.H.M., Mendling, J.: Configurable multi-perspective business process models. Information Systems 36(2), 313–340 (2011)

Control Automation to Reduce Costs of Control

Rob Christiaanse[1] and Joris Hulstijn[2]

[1] VU University and EFCO Solutions, Amsterdam
[2] Delft University of Technology, Delft
r.christiaanse@efco-solutions.nl, j.hulstijn@tudelft.nl

Abstract. Much compliance effort concerns adherence to contracts. Parties to a contract need to make sure the other party fulfills the contract. To this end they may require additional controls in the business process. Controls have costs. In this paper we argue that fully automated controls help to lower control costs, because (i) the can help prevent misstatements (compliance by design) or (ii) they increase the quality of audit evidence and thereby reduce the audit risk and additional audit fees. The line of reasoning is illustrated by a case study of the implementation of automated controls on the procurement process for public transport services for the elderly and disabled. The case study suggests open issues, which can be linked to concepts from Normative Multi Agent Systems.

Keywords: compliance monitoring, auditing, evidence.

1 Introduction

Business reality consists of contractual arrangements between actors, like seller and buyer. A contract is a statement of intent to regulate behaviour. In this sense, "… most organizations are simply legal fictions which serve as a nexus for a set of contracting relationships among individuals" [17] (p.310). Within a larger rule-governed setting a contract provides for a comprehensive set of both constitutive and regulative rules [23]. Constitutive rules introduce abstract classifications of facts and entities and by describing the (legal) consequences of actions, whereas regulative norms provide conditional obligations with an associated sanction or reward [4]. Such provision can be fruitfully studied using Normative Multiagent Systems [7].

Businesses put more and more effort into collecting evidence to demonstrate compliance with contracts. Much of the (corporate) governance debate concentrates on the key question: "What constitutes an efficient control system?" [24, 27, 29]. One way to deal with increasing administrative burden, i.e., the cost of control, is to use information technology in a clever fashion. Generally, information systems may help (i) to collect and possibly even analyse evidence, and (ii) to facilitate the organization to be 'in control' by preventing undesired behaviour. This may be called 'compliance by design'[22]. The term was initially used in the context of business process management and is related to conformance checking [21]. Conformance checking assumes there is a reference model (de jure model) with process constraints against which the evidence of process behaviour (de facto model) can be verified. But is is no

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 322–336, 2012.

specified how to derive the reference model, e.g. from legal sources, technical standards or best practices. Also, given a de facto model, how the raw evidence needs to be interpreted and mapped onto the de jure model is left unspecified. Issues due to transaction costs, auditing roles and responsibilities and the meaning of evidence are not sufficiently addressed. We therefore prefer to use the term 'compliance by design' in a broader sense, referring to an integrated design of organizational, procedural and technical measures, to make sure the organization is evidently compliant.

In this paper we analyse the problem of assurance provision within a fully automated environment, focusing on the strength of evidence and the roles of management, auditors and stakeholders in setting up a control system to minimize the compliance control costs. Our research question is twofold:

(1) How can organizations ensure and prove to others that they are keeping the contract, while minimizing the costs of control?

Our approach will be to analyse a real world case study. The case study concerns the set-up of automated control measures for ensuring compliance of the monthly invoice with the contract, regulating public transport services for the elderly and disabled.

To analyse the case study we will use techniques from the field of Normative Multi Agent Systems (NMAS) [4, 7]. We believe NMAS concepts bring out observations, which may bridge the gap between abstract norms and systems specifications.

The remainder of the paper is structured as follows. In Section 2 we discuss the various aspects of control and the role of collecting strong evidence. In Section 3 we describe the case study, illustrating the difficulties of a highly regulated application domain. In Section 4 we provide some open issues for research, indicating how Normative Multi-Agents Systems could be used to analyse these issues.

2 Control, Business Compliance and Assurance

So what are governance and control? Governance is related to delegation. The owner of a firm must delegate tasks to its members. Delegating tasks raises specific control problems for the principal, in particular the problem of private information [19]. Private information problems can be of two types, namely moral hazard (hidden action) and adverse selection (hidden knowledge). A third type of problem concerns non-verifiability. Non-verifiability of information occurs when a principal and an agent share information ex-post which cannot be observed by a third party, who may also have an interest in the outcomes, for example a regulator.

These control problems are generally countered by additional control measures, like supervision, procedures, verification, or automated application controls, etc. Controls come with a cost. The cost of control depends on two factors: (1) the incremental monetary cost of the control instrument chosen, including opportunity costs and (2) the cost of negative side effects caused by or related to the controls [20]. In particular, there are four types of problems: (2.1) behavioural displacement (alter behaviour to avoid the controls; (2.2) gamesmanship (exploit the rules to individual or organizational advantage, e.g. fully use subsidies even when not needed

(2.3) operating delays (additional controls like supervision or signing off slow down or disrupt primary processes); and (2.4) negative attitudes (controls may highlight extrinsic motivation, e.g. targets, rather than intrinsic motivation, which has a negative effect on motivation of employees).

How can we measure the costs of controls? In the case of controls being added to contracts, one way to make sense of the costs is to use Transaction Cost Economics (TCE) [29]. In TCE costs are defined as "The *ex ante* costs of searching, drafting, negotiating and safeguarding an agreement and, more especially, the *ex post* costs of maladaptation and adjustment that arise when contract execution is misaligned as a result of gaps, errors, omissions, and unanticipated disturbances" [26]. A rational actor will prefer the governance structure with the lowest transaction costs [11, 29]. Optimal control is achieved, when the expected control losses are smaller than the cost of implementing more controls [20, 24]. From a management perspective there are several strategies to deal with control risks: (1) activity elimination, (2) centralization, (3) risk-sharing, (4) automation and (5) mitigation. Options (1), (2) and (3) are strategic issues; usually these are given. Therefore we will focus on (4) automation and (5) mitigation.

A mayor source of administrative burden is due to errors and mistakes in the transaction data, and in general, lack of information quality. Traditionally these problems are dealt with by the mitigation strategy, reducing errors by additional human checks. Such action controls, like workflow rules, direct supervision, separation of duties, procedures, etc., are rather costly. Another way to deal with the effects of inaccuracy, inconsistency and lack of motivation is to use information technology in a clever fashion. After all, software is never dishonest or disloyal [20]. Built-in controls provide reasonable assurance that performed activities meet process specifications and contracts are kept against acceptable control costs.

The private information problem of agency theory is in practise dealt with by auditors. Auditors are independent experts, monitoring the firm on behalf of stakeholders of the firm. For decision purposes, stakeholders need reliable financial information about the status of an organisation and its performance [12, 18]. Traditionally auditors are responsible for providing reasonable assurance that (financial) information is free from material misstatements [16, 18]. In this respect materiality reflects "... the magnitude of an omission or misstatement of (financial) information that, inferred in the light of surrounding circumstances, makes it probable that the judgment of a reasonable person relying on this (financial) information would have been changed or influenced by the omission or misstatement" [16]. In general, auditors use the audit risk model in planning the audit process: "… the audit risk concerns the risk that the auditor may unknowingly fail to appropriately modify that his or her opinion on (financial) information that are materially misstated" [18].

(2) *Audit Risk = Inherent Risk * Control Risk * Detection Risk*

The audit risk model should be thought of as an aid in understanding how various factors affect the amount of substantive testing required. Inherent risk is defined as the a priori likelihood that a management assertion may be misstated, so before considering the possible effect of (internal) controls. Control risk is defined as the likelihood that (internal) controls will not prevent or detect a misstatement. Detection risk refers to the risk that an auditor will not detect a material misstatement, and consequently fail to appropriately modify his or her opinion. Detection risks concern failures of substantive testing and the appropriateness of audit evidence, including the strength or persuasiveness of audit evidence.

In general, there are six ways of obtaining audit evidence: (1) Inspection, (2) Confirmation, (3) Observation, (4) Re-performance, (5) Analytical evidence an (6) Client inquiry [18]. Inspection can be subdivided into inspection of tangible assets and inspection of documents and records. There exist a hierarchy of evidence reliability. Types (1), (2) an (4) are considered the best forms, while types (3), (5) and (6) are considered good or weak, depending on the sources (human or automated), expectations, procedures and planning of the client inquiry. Types (1), (2) and (4) are also the most time consuming for the auditor and therefore the most expensive for the client. So there is a trade-off between the quality of evidence and the costs of control.

How can we conceptualize automated controls? Following Clark and Wilson [10] we distinguish the notions of external consistency and internal consistency of the data produced by a system. Suppose we have a well-managed computer system. Its specifications have been verified to be correct, and the system itself has been tested and behaves according to its specifications. That means that when we enter data into the system that is valid, valid data will ensue (internal consistency). However, even in such a near-perfect system there is nothing to ensure external consistency: correspondence with reality. In general, external consistency can only be ensured by a combination of organizational measures (segregation of duties), procedural measures (e.g. maintenance, supervision) and physical measures (e.g. gates, use of IDs). These are called indispensable measures.

Built-in automated controls may have two effects. Clearly, built-in controls help to prevent material misstatements so they reduce the internal control risk. Built-in controls may also mean that audit evidence will be stronger, because of the availability of reliable transaction data. If this is the case, the detection risk will become lower: the probability that the auditor would miss a possible material misstatement is reduced. Therefore, for both these reasons, in principle fewer substantive tests are needed. We summarize this expectation as follows:

(3) Control automation reduces the costs of control, by increasing both
  a. the probability of preventing misstatements, and
  b. by aiding the auditor improve the audit by strengthening the audit evidence.

In the following section we will explore this double expectation in practice.

3 The Case of Procurement for Public Transport Services

We expect that control automation will enhance the control effectiveness and quality of evidence. We investigate this claim by analysis of a real world case.

3.1 Case Description

The case concerns the public procurement process for care-related public trans- port services. In the Netherlands municipalities have the responsibility to procure public transport services, facilitating in particular the elderly, people with functional disabilities and people with mental health problems. As you can imagine this is a highly regulated sector. Relevant provisions are found for instance in European Directive 2004/18/EC about public transport contracts, ensuring the principle of equal treatment, the principle of proportionality and the principle of transparency.

We focus on one particular contract: TaxiBus (TED 2010/S 31-044711 dated 13/02/2010). The case is exemplary for a large class of care-related services. The contract involves the following participants.

The contracting authority is called SRE, procuring the transport services and paying the invoices. SRE represents fifteen municipalities in the Eindhoven region, in the role of stakeholders. SRE is assisted by an external auditor.

The transport service provider (TSP) provides care-related ‘taxi bus’ services on a demand basis. It receives a monthly fee from SRE as well as individual contributions from (some) passengers. TSP keeps a record of all trips being requested, executed and cancelled, including data about patients and other travellers.

3.2 Contract

The contract contains provisions about safety, competence of the drivers, and quality. Other requirements concern the contract area, tariff rules based on the number of travel zones, and the types of clients (residents; elderly; companions). In addition to a fee from SRE, passengers must also pay a contribution. Pensioners pay a lower contribution. The contract contains performance provisions about delays or maximum driving time. When TSP does not meet these criteria a penalty must be paid.

1. *The carrier receives a fee based on the number of transportation zones covered in a trip, based on a rate per area. No additional embarkation zone is covered.*
   a. *The carrier charges the fee per area. The carrier charges the net amount of travelled trips, excluding received contributions;*
   b. *All amounts are exclusive of VAT.*
   c. *The carrier may not charge SRE an additional fee, for:*
      - *a medical supervisor (see TED 2010/S 31-044711 par. 1.6.4)*
      - *a companion (see TED 2010/S 31-044711 par 1.6.4); in this case the contribution is collected by the driver;*
      - *children up to four years old;*
      - *guide-dogs or other assistance dogs, recognizable as such.*

      The carrier may charge trips in addition to the contribution, in case the passenger and his companions are family;
   d. *The following trips are not reimbursable:*
      - *transport of animals, goods etc;*
      - *trips that do not meet the registration requirements, as set out in TED 2010/S 31-044711 par. 1.17.1 (except GPS-coordinates);*

- *cancelled trips;*
- *no-show rides.*

2. *Passenger rate per zone. A passenger pays a contribution based on the number of transportation zones, plus an embarkation zone.*

Every month SRE receives an invoice. How can SRE establish the accuracy and legitimacy of the invoice? By contract, the invoice is accompanied by an Excel data file, containing the grounds for billing: the registration of the trips being ordered, executed, and cancelled. For example, patient IDs, scheduled and actual departure and arrival times, locations and so on. The contract contains a data protocol, specifying which data elements the Excel file should contain and in which order.

3. *The carrier maintains tamper-resistant automated records of the number of requested and conducted trips, organized in runs based on the route planning software in the central office and on the trip registration device in each vehicle. This means that refused, not implemented or no-show rides are recorded.*
4. *The carrier shall provide the client with a monthly Excel file, detailed as specified in 5, containing all requested and actually implemented customer trips. This means that refused, not implemented or no-show rides are included.*
5. *For each one-way trip, at least the following data is registered (selection):*
   - *unique trip identification number;*
   - *identification code of any subcontractor who performs the trip;*
   - *transit pass number (customer identification number), if any, by means of which user and residential municipality can be identified;*
   - *municipality where the customer resides;*
   - *customer type(WMO customer, elderly customer or regular public transport);*
   - *agreed departure time;*
   - *agreed arrival time;*
   - *actual departure time;*
   - *actual arrival time;*
   - *travel duration in minutes;*
   - *departure address;*
   - *destination address;*
   - *departure and destination public transportation zone;*
   - *length of the trip through some specific areas (excluding pick-up zone);*
   - *length of the trip mileage;*
   - *number of zones to be charged to contracting authority*
   - *number of persons to be transported;*
   - *type accompanying person (medical supervisor, social companion, non-paying children, family taxi, etc.);*
   - *travellers contribution (including any companions' contribution);*
   - *required equipment and aids (wheelchairs, scooters, guide dogs, etc.);*
   - *status of the ride (out, refused, not implemented, no-show, implemeted);*

This trip administration forms the ground for billing, to be provided to both SRE and the auditor, so that the correctness and completeness of the invoice can be ascertained.

The Excel file is however being provided by the transport service provider. So the contract stipulates the need for assurance, obtained from an external auditor.

6. *“Each year the service provider is required, no later than three months after the end of a calendar year, to present an audit opinion asserting the correctness of the invoices, the delivered data and the management information. This audit opinion is accompanied with a report of findings concerning the comprehensiveness of the accounting information system. The auditing fees are paid by the service provider” (TED 2010/S 31-044711, p 32);*

3.3 Negotiation about Controls

SRE management has to decide what controls are requested to prevent misstatements in the monthly invoices. However, these controls have to be paid for, by tax payers’ money. There is a trade-off between the effectiveness and costs of control. This results in a dialogue between the municipality in the role of stakeholder, and SRE management in the role of contracting authority. Note that such dialogues are common in audit dialogues about the design of a system of controls [9].

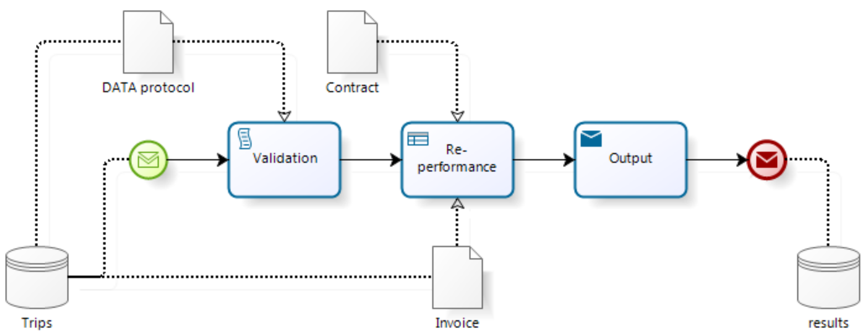


Fig. 1. General overview of establishing accuracy of the monthly invoice

SRE argues that, if TSP is in control of its billing process, the invoice and the data provided in Excel will be accurate. The stakeholder might ask: “How can you be sure?” SRE will respond that the contract obliges TSP “to present an audit opinion asserting the correctness of the invoices, the delivered data and the management information.” Now the stakeholder might argue: “Do we not pay too much, because we pay in advance, and when an error is detected by TSP’s auditor, it will not lead to any correction of the invoiced amounts, for the detected error will remain within the error threshold?”. After some pondering: “No, it seems to me that the invoices are simply illegitimate in case the ground for billing is not correctly established for each trip”[1]. SRE has foreseen this line of reasoning and states: “That is why we receive on a monthly basis an Excel file containing all requested and actually implemented trips.

[1] Law on audit of counties and municipalities, 2003.

Plus we have a data protocol in which data definitions are defined". The stakeholder responds positively: "So, you are able to verify the grounds for billing using the Excel file accompanying the invoice?" SRE answers positively but claims they do not have the resources to manually check all trips (50.000 per month). The solution for the control problem is therefore full automation, i.e., audit evidence type (4).

3.4 Control System Specification

Figure 1 contains a possible architecture for establishing accuracy of the monthly invoice on the basis of an independent evidence file about the trips being conducted. Here we will give semi-formal characterisation of the requirements. The validation procedure is written in PHP. For representing contracts there are better options, such as FCL [14]. Here, we merely want to characterize the kinds of reasoning about the evidence. We will use Predicate Logic notation.

Suppose a contract is a set of clauses $C = \{C_1, \ldots, C_n\}$, where each clause may contain free variables referring to trips, passengers, and other data types. We have a set E of facts about trips. Facts are instantiated and do not contain free variables. Invoice amount x is based on price function '$f(.)$' over trips. ε is an acceptable margin of error (materiality). Now we can formulate the control requirements as follows:

(6)	$E \models \forall t(C_1(t) \wedge \ldots \wedge C_n(t))$	legitimacy
(7)	Let $y = \Sigma t\, f(t)$, for all t such that $E \models C_1(t) \wedge \ldots \wedge C_n(t)$	recalculation
(8)	Invoice x is accurate when $y - \varepsilon < x < y + \varepsilon$.	accuracy

Concerning evidence E, two major issues must be addressed: internal integrity and external integrity [10]. The internal integrity of data refers to conformance of the data to the data protocol. The external integrity depends on the authorization level of the recorded data, and on the presence of segregation of duties and other input controls during generation of the evidence E (indispensable control measures) [2]. Evidence about these internal controls is represented in an audit assertion IC, obtained once a year from the external auditor. We suppose the auditor uses a norm N about adequacy of internal controls in a billing process. If needed, the auditor can also verify registered trips against the system's planning module; data about implemented trips is obtained from the taxi drivers' mileage administration; refused and no-show trips can be obtained from passenger day schedules.

However external integrity of the data can also (partly) be established on the basis of the data itself. After all, taken together, the data must form a set of well-formed transactions. This can be verified by so called reconciliation relations, based on accounting theory [25]. Both syntactic constraints and reconciliation relations are specified in a testing procedure M.

In our experience, there are at least four stages in a validation procedure M. Stage 0 validates whether the evidence file contains the attributes defined in the data protocol. Stage 1 validates the syntax. Stage 2 validates whether the content of an attribute (if defined) conforms to the value constraints (semantics). Stage 3 validates relation constraints, in particular whether two or more attributes (if defined) taken together form a well-formed transaction.

So, reliability of evidence E is established by comparing evidence of the internal controls IC used in generating E to a norm N (external auditor), and by automatically validating E on the basis of checks and recalculation specified in M.

(9)	assert(auditor, IC meets N)	internal controls meet norm N
(10)	validate(E,M)	automated validation test M

This will work, when M is sound, i.e., when validation test procedure M is strong enough to imply the constraints in the contract.

(11) if validate(E,M), then $E \models \forall t(\, C_1(t) \wedge \ldots \wedge C_n(t)\,)$ soundness

It is important that all stakeholders can verify soundness of M. To facilitate this, the mapping constraints are written in PHP and the validation code is made public.

4 Organizational Roles in Automated Audit Facilities

Control and auditing is in essence about setting, following and verifying norms. In the case study we have seen that actors are negotiating about the controls to be included in a contract. Here we are interested in the assurance question: who gets to decide what 'counts as' evidence? Who is verifying that evidence? Who will pay for such assurance services? Such a negotiation suggests a game theoretic setting, because actors reason about what they should do in response to what the other party is likely to do, under condition of a set of controls. In this section we will use ideas from qualitative game theory expressed in Normative Multi-agent Systems [5, 7], to generalize the case study and analyse the roles in assurance provisioning.

Normative multiagent systems are systems composed of autonomous agents with a normative system, i.e. a set of norms (obligations and conventions), to regulate behaviour of the agents. Here we will only use NMAS terminology as a conceptual tool to make an analysis of the stakeholders and their relationships. The particular version we are inspired by [7], makes use of input-output logic, but that is totally irrelevant to the point of the paper. Obviously, there are much better formalisms to express contractual clauses in logic, such as versions of RuleML [13], or to translate contracts into constraints on business processes, such as Formal Contract Logic (FCL) [14] based on defeasible logic [1]. However, here we are not interested in the logical properties or the representation question; the point is agenthood: who decides?

Consider an abstract transaction as depicted in Figure 2. It shows various embedded legal or professional jurisdictions, depicted as spheres. The kernel is made up by the contractual obligations of buyer *b* and service provider *s*. Both parties can hire an auditor, *ab* and *as* respectively, also regulated by contracts. In particular, *ab* provides assurance over service delivery, based on evidence provided by *s*. Auditor *as* provides an audit assertion about the internal control system of *s*, so indirectly about the way the evidence is being generated. Moreover, *ab* may take a mediating role in contract negotiations, providing advice about the evidence requirements. Note that auditors have shared professional standards and values about what counts as proper evidence. There are also domain specific provisions (e.g. about health care and public transport) and all of this is set within national or international contract law.

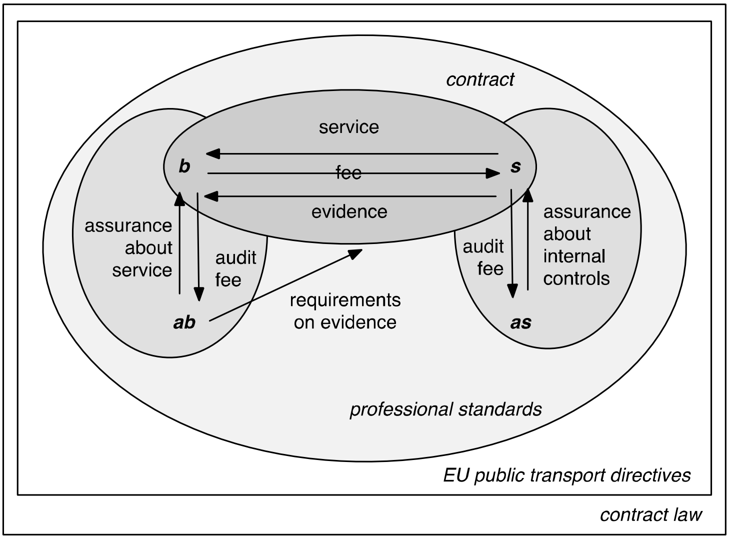


Fig. 2. Legal setting of contract negotiations

As is clear from the picture, we need constitutive rules, based on the distinction between constitutive and regulative rules popularized by Searle [23]. Roughly, constitutive rules help to shape social reality, just like the rules of chess constitute the game of chess. Regulative rules attempt to alter the behaviour of agents within the normative system, by qualifying certain behaviour as violations and making it subject to sanctions. Constitutive rules are generally of the form: "such and such an *X* counts as *Y* in context *S*" where *X* is any object, fact or event and *Y* is a label that qualifies *X* as being of a certain category, often a social or institutional fact. In our case, validation procedure *M counts as* an acceptable test for an accurate and legitimate invoice. The context *S* is interpreted as a set of embedded institutional jurisdictions, as in Figure 2. *S* is common knowledge .For an elaborate analysis of counts-as see [15].

In our simplified NMAS approach, agents are just sets of sets of production rules, with a preference order $<a$. Rules are of the form $p1, \ldots, p\mathrm{n} \rightarrow q,$ where all pi and q are literals: facts or negations of facts. When $n = 0$, we just have a fact, written $\rightarrow q$. Rules are clustered in sets: belief rules Ba and goal rules Ga. To calculate the effect of reasoning with these sets of rules, we need a consequence relation. This is provided by taking the fixed point, where rules are applied in an order specified by a preference relation $<a$. Decisions are consistent sets of outcomes of the consequence relation.

Now we move to the representation of an obligation. Following Boella and van der Torre [7] obligations are analysed in terms of goals of the addressee of the norm, agent *a,* and of the institution or normative agent ***n***. The idea is that an obligation is a goal of the normative agent ***n***, according to the slogan: "Your wish is my command".

There are six clauses to the analysis [3]. The first clause states that an obligation to achieve or maintain *p* is the goal of the normative agent to achieve or maintain *p*. The second and third clauses deal with violations. Obligations only become visible through defining under which conditions they are not met. The third clause says that in principle, the agent desires no violations. The fourth and fifth clauses deal with sanctions. Normative agent **n** prefers not to count behaviour as a violation, and does not want to apply sanctions unless there is a violation; otherwise sanctions would be arbitrary. Finally, the agent really doesn't like the sanction, or it would have no deterring effect.

Definition (Obligation): Agent a is obliged towards normative agent **n** to decide to do x, in case of condition Y, against sanction s, in context of institution S, written $S \models \mathrm{Obl}(a, \boldsymbol{n}, x, s, Y)$, if and only if:

(1) $Y \rightarrow x \in \mathrm{G}a$	agent a prefers to oblige
(2) $Y \cup \{\neg x\} \rightarrow \mathrm{V}(\neg x, a) \in \mathrm{G}\boldsymbol{n}$	***n*** prefers to signal violations, if needed
(3) $\rightarrow \neg\mathrm{V}(\neg x, a) \in \mathrm{G}\boldsymbol{n}$	***n*** prefers not to recognize a violation
(4) $Y \cup \{\mathrm{V}(\neg x, a)\} \rightarrow s \in \mathrm{G}\boldsymbol{n}$	***n*** prefers to sanction, if needed
(5) $\rightarrow \neg s \in \mathrm{G}\boldsymbol{n}$	***n*** prefers not to sanction a arbitrarily
(6) $\rightarrow \neg s \in \mathrm{G}a$	agent a prefers no sanction

Now lets suppose an agent has a private incentive to violate the obligation, for example, to behave opportunistically. Such an incentive can be represented by an additional clause, specifying that (apparently) violations are preferable.

(7) $\rightarrow \neg x \in \mathrm{G}a$

Now we make several observations. First, this scheme is only going to work, if agent a prefers to comply, i.e. (7) $<a$ (1), or else if the deterring effect of sanctions is larger than the gain of a violation, so (7) $<a$ (6). The normative agent must also make trade-offs concerning efforts. In the context of such trade-offs, detecting violations should have priority: (3) $<\boldsymbol{n}$ (2). Similarly, in the trade-off about applying sanctions, the normative agent should prefer to apply them when necessary: (5) $<\boldsymbol{n}$ (4).

Given that some of these priorities are set, clause (2) is crucial: it specifies violation conditions. However, it could be the case that $\neg x$ cannot be directly observed. Which agent is authorized to specify what counts as a violation? That is essentially a constitutive question. Boella and van der Torre make the following simplifying move: constitutive rules are modelled by the beliefs of the normative agent ***n***. The normative agent further determines what further detection conditions Z count as sufficient evidence of a violation, i.e. $\neg x$.

(2') $Y \cup \{\neg x\} \rightarrow \mathrm{V}(\neg x, a) \in \mathrm{B}\boldsymbol{n}$
(8) $Z \rightarrow \neg x \in \mathrm{B}\boldsymbol{n}$

In a preference order for a rational agent, belief-rules rules outrank goal-rules, to avoid wishful thinking [8]. By putting clause (2') into the beliefs, the normative agent must detect a violation when the conditions are met. Clause (8) provides further detection conditions Z. A separate clause is logically not necessary, but in practice non-compliance can show in many different ways (see example below).

4.1 Automated Control

Now we move back to analyse the case study. Buyer *b* has an obligation to pay for delivery, with a sanction of discontinuation of the services. Service provider s has an obligation to deliver upon payment, with a sanction of a fine.

Obl(*b*, ***n***, *pay, stop deliver,* {*deliver*})
Obl(*s*, ***n***, *deliver, fine,* {*pay*})

Now we focus on clause (2') and clause (8). In terms of the case study, the buyer (SRE) has paid in advance on a monthly basis, but needs to verify delivery.

(2') $\{pay\} \cup \{\neg deliver\} \rightarrow \mathrm{V}(\neg deliver, s) \in \mathrm{B}\boldsymbol{n}$

First, what evidence counts as non-delivery? These three conditions have been discussed in the case. In the context of contract TED 2010/S 31-044711, the following conditions count as evidence of non-delivery: no qualified audit opinion has been provided about the internal controls *IC,* evidence *E* has not been validated against mapping *M*, or invoice *x* differs more than an error threshold from the recalculation.

(8) $\{\neg \exists as\ (assert\ (as, IC\ meets\ N))\ \} \rightarrow \neg deliver \ \in \mathrm{B}\boldsymbol{n},$
$\{\neg\ validate\ (E,M)\ \} \rightarrow \neg deliver \ \in \mathrm{B}\boldsymbol{n},$
$\{\ \Sigma t\ f(t) + \varepsilon < \ x \text{ , for } t \textit{ such that } E \models C_1(t) \wedge \ldots \wedge C_n(t)\ \} \rightarrow \neg deliver \ \in \mathrm{B}\boldsymbol{n}$

Second, which actor plays the role of the normative system ***n***? We have a number of choices in modelling the situation.

Choice A. The normative system ***n*** corresponds to the legal authority, which maintains contract law, in our case, the Dutch legal system, with additional provision from EU directives. That means that formally, obligations of buyer and seller are directed towards a legal abstraction; the fact that they deal with each other can be expressed as part of the contents of the contract and violation conditions *Y.*

Choice B. The normative system corresponds to the other contractual party. That means that both formally and in fact obligations are directed towards each other. It also means that parties themselves recognize violations and apply sanctions.

In the case study, the contractual parties do in fact recognize violations themselves, on the basis of what is considered reliable evidence set by an external auditor, and also the contractual parties themselves are allowed to apply sanctions. Therefore we choose option A. Alternatively, we can distinguish different kinds of ***n*****,** performing different tasks. For instance, we can distinguish a detector agent, performing clause (2) – in the case the buyer's auditor – and a sanctioning agent – in the case the opposite contractual party. For similar ideas of separating the roles of agents see [6]. As we have seen, the contract itself stipulates the use of external evidence of an auditor. The auditor is professionally bound by the norms of the profession and by the general legal setting of Dutch contract law and European Directives. Each of these bounds can again be modelled as an obligation, with corresponding violation clauses (2), (2') and (8) which now stipulate evidence of meta-violation.

5 Conclusions

What is considered an efficient control system? How can we reduce the costs of control? In this paper we have explored the possibilities of automated controls.

The main proposition, that we generated based on agency theory and audit theory, is that automated controls can reduce the costs of control, for two reasons. First, automated controls prevent misstatements and limit the internal control risk. Second, automated controls increase the quality of evidence, and thereby decrease the detection risk. Given a fixed audit risk, the auditor therefore has to perform much less substantive testing. Another benefit is that assurance can be provided when needed.

The role of the auditor is to set standards for evidence (data protocol) and to set up automated validation and verification techniques. An auditor (can be another one) must also verify the irreplaceable controls (internal control, procedures and reliability of the computer system), in order to guarantee external consistency. The main work in verifying accuracy and legitimacy is done by verifying the well-formedness of transactions through reconciliation and recalculation. So in general internal consistency can help to bootstrap external consistency, given data interdependencies and a little circumstantial evidence about reliability of the computer system

The ideas expressed above have been exemplified by a detailed case study about automated controls for verifying compliance to a contract concerning public transport services for the elderly and disabled. This constitutes a heavily regulated domain. The case shows that it is possible to use a monthly data file, provided by the service provider, as evidence of accuracy and legitimacy of the monthly invoice. The data file can be verified automatically. The data is so much interdependent that manipulation will show up in the verification process, unless the provider would deliberately use audit software to manipulate the data. In addition, the software and the billing process which generate the data file are audited once a year.

In the example case, validation is done on a monthly basis, in batch mode. However, the same kind of validation could in principle also be done per row on a dynamic basis. This is a form of continuous monitoring or continuous assurance [28]. Recalculation requires totals, but in principle also recalculation could be done dynamically for any total (per day, per week, per month) as the process develops.

Parties negotiate about the required controls, because controls come at a cost. The whole setting is therefore fruitfully analysed using techniques borrowed from Normative MultiAgent Systems. Crucial is the distinction between regulative rules and constitutive rules, which help to specify what counts as evidence of a violation. Summarizing we can say that the NMAS perspective has forced us to be explicit about (i) who is authorized to set standards of evidence (i.e. the auditor), and (ii) how this evidence provisioning is regulated (i.e., by the data protocol in the contract, and the automated verification procedure) The analysis suggests a nested set of NMASs, regulating the contract at the core, but also the contracts of the auditors involved, embedded in various legal and professional jurisdictions.

References

1. Antoniou, G., et al.: Representation results for defeasible logic. ACM Transactions on Computational Logic 2(2), 255–287 (2001)
2. Blokdijk, J.H.: Tests of control in the audit risk model: effective? Efficient? International Journal of Auditing 8, 185–194 (2004)
3. Boella, G., Lesmo, L.: A game theoretic approach to norms. Cognitive Science Quarterly 2(3-4), 492–512 (2002)
4. Boella, G., van der Torre, L.: Regulative and Constitutive Norms in Normative Multiagent Systems. In: Procs. of 9th International Conference on the Principles of Knowledge Representation and Reasoning (2004)
5. Boella, G., van der Torre, L.: Contracts as Legal Institutions in Organizations of Autonomous Agents. In: Proceedings of the Third International Joint Conference on Autonomous, Agents and Multi Agent Systems, AAMAS 2004 (2004)
6. Boella, G., Hulstijn, J., van der Torre, L.: Virtual Organizations as Normative Multiagent Systems. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences, HICSS 2005, p. 192 (2005)
7. Boella, G., van der Torre, L.: A game theoretic approach to contracts in multiagent systems. IEEE Transactions on Systems, Man and CyBernetics - Part C 36(1), 68–79 (2006)
8. Broersen, J., et al.: Goal Generation in the BOID Architecture. Cognitive Science Quarterly 2(3-4), 431–450 (2002)
9. Burgemeestre, B., Hulstijn, J., Tan, Y.-H.: Value-based Argumentation for Justifying Compliance. Artificial Intelligence and Law 19(2-3), 149–186 (2011)
10. Clark, D.D., Wilson, D.R.: A Comparison of Commercial and Military Computer Security Policies. In: IEEE Symposium on Security and Privacy (1987)
11. Dekker, H.C.: Control of Inter-Organizational Relationships: Evidence on Appropriation Concerns and Cordnation Requirements. Accounting, Organisations and Society 29, 27–49 (2004)
12. FASB, Statement of Financial Accounting Concepts No. 2: Qualitative Characteristics of Accounting Information, Financial Accounting Standards Board (1980)
13. Governatori, G., Rotolo, A.: Modelling Contracts Using RuleML. In: Gordon, T. (ed.) Legal Knowledge and Information Systems (Jurix 2004), pp. 141–150. IOS Press, Amsterdam (2004)
14. Governatori, G., Rotolo, A.: Norm Compliance in Business Process Modeling. In: Dean, M., Hall, J., Rotolo, A., Tabet, S. (eds.) RuleML 2010. LNCS, vol. 6403, pp. 194–209. Springer, Heidelberg (2010)
15. Grossi, D., Dignum, F.P.M., Royakkers, L.M.M., Meyer, J.-J.C.: Collective Obligations and Agents: Who Gets the Blame? In: Lomuscio, A., Nute, D. (eds.) DEON 2004. LNCS (LNAI), vol. 3065, pp. 129–145. Springer, Heidelberg (2004)
16. Houston, R.W., Peters, M.F., Pratt, J.H.: The Audit Risk Model, Business Risk and Audit-Planning Decisions. The Accounting Review 74(3), 281–298 (1999)
17. Jensen, M.C., Meckling, W.H.: Theory of the Firm: Managerial Behavior, Agency Costs and Ownership Structure. Journal of Financial Economics 3(4), 305–360 (1979)
18. Knechel, W., Salterio, S., Ballou, B.: Auditing: Assurance and Risk, 3rd edn. Thomson Learning, Cincinatti (2007)
19. Laffont, J.-J., Martimort, D.: The Theory of Incentives: the Principal-Agent Model. Princeton University Press (2002)

20. Merchant, K.A.: Modern Management Control Systems, text & cases. Prentice Hall, New Jersey (1998)
21. Rozinat, A., van der Aalst, W.M.P.: Conformance checking of processes based on monitoring real behavior. Information Systems 33(1), 64–95 (2008)
22. Sadiq, S., Governatori, G.: The journey to business process compliance. In: Cardoso, J., van der Aalst, W.M.P. (eds.) Handbook of Research on Business Process Management, pp. 426–454. IGI Global (2009)
23. Searle, J.R.: The Construction of Social Reality. The Free Press (1995)
24. Speklé, R.F.: Explaining management control structure variety: a transaction cost economics perspective. Accounting, Organizations and Society 26(4-5), 419–441 (2001)
25. Starreveld, R.W., de Mare, B., Joels, E.: Bestuurlijke Informatieverzorging, vol. 1. Samsom, Alphen aan den Rijn (1994) (in Dutch)
26. Swedberg, R.: Economic versus sociological approaches to organization theory. In: Tsoukas, H., Knudsen, C. (eds.) Oxford Handbook of Organization Theory, Meta-Theoretical Perspectives. Oxford University Press, Oxford (2003)
27. Tirole, J.: Corporate Governance. Econometrica 69(1), 1–35 (2001)
28. Vasarhelyi, M.A., Alles, M., Kogan, A.: Principles of analytic monitoring for continuous assurance. Journal of Emerging Technologies in Accounting 1(1), 1–21 (2004)
29. Williamson, O.E.: Transaction Cost Economics: The governance of contractual relations. Journal of Law and Economics 22, 3–61 (1979)

Introducing a Mashup-Based Approach for Design-Time Compliance Checking in Business Processes*

Cristina Cabanillas, Manuel Resinas, and Antonio Ruiz-Cortés

Universidad de Sevilla, Spain
{cristinacabanillas, resinas, aruiz}@us.es

Abstract. Business process compliance tries to ensure the business processes used in an organization are designed and executed according to the rules that govern the company. However, the nature of rules (expressed in natural language) and the large amount of elements that can be involved in them make their materialization and automated checking quite difficult. That is why the existing support for compliance checking is generally restricted to specific kinds of rules (e.g. rules affecting the control flow of the process). In this paper, we introduce *compliance mashups*, and show how a mashup-based approach can help solve the problem of rule specification and checking at design time. Some advantages of such an approach are that: (i) any kind of rule can be specified, which implies that each user can specify a rule according to his/her interpretation of the rule; (ii) building the compliance mashup is transparent to the formalism(s) used to implement it, so different techniques can be used together; and (iv) mashup components or parts of them can be re-used. As an example we use this approach to build mashups to specify and check rules related to human resource management in business processes at design time.

Keywords: Business process compliance, rule specification, compliance mashup, natural language disambiguation, design-time compliance checking.

1 Introduction

Business process (BP) compliance consists of ensuring the BPs used in an organization are designed and executed according to the policies that govern the company. Policies can be decomposed into rules that introduce constraints relating to different aspects of the BPs, such as the execution order of the activities (i.e. control flow), the data accessed and managed, or the people (a.k.a *human resources* or just *resources*) that participate in the process.

* This work has been partially supported by the European Commission (FEDER), Spanish Government under the CICYT project SETI (TIN2009-07366); and projects THEOS (TIC-5906) and ISABEL (P07-TIC-2533) funded by the Andalusian Local Government.

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 337–350, 2012.

Compliance can be checked at different phases of the BP lifecycle [1], which results in two big compliance checking modalities [2]. The most proactive way to check compliance is Forward Compliance Checking (FCC), which can be divided into two sub-approaches: Design-Time Compliance Checking (DTCC) and Run-Time Compliance Checking (RTCC). DTCC is usually performed after BP modelling with the aim of ensuring that the process is compliant with the given rules before its execution, thus saving time and effort to business analysts. An example is tool OPAL developed by some researchers from IBM Research China, which uses model-checking techniques to automatically verify control flow-related rules in BP models [3]. Nevertheless, there is a bunch of proposals to deal with DTCC based on diverse techniques. A common feature of most of them is that they rely on annotated BP models to check compliance [4,5]. RTCC techniques check rules at run time, so if a rule is violated or some problematic situation arises while running the process we might be able to solve the problem on time to avoid ending in a non-compliant state. This requires some kind of software for business activity monitoring (BAM) in BP management systems (BPMS). European Project COMPAS worked in this direction [6]. Finally, Backward Compliance Checking (BCC) focuses on determining whether past instances of a BP were compliant with rules from information stored in history logs. The result of BCC helps stakeholders to be prepared for audits. ProM [7], a tool for process mining, contains plugins to perform this kind of compliance checking, mostly focused on control flow issues [8]. A summary of features that should be considered when developing a BP compliance management system (BPCMS) that covers all the phases of the BP life cycle was introduced in [9].

In this paper we focus on FCC, particularly on DTCC. In DTCC, rules are checked against BP models, which involves translating them into a formal language that can be automatically processed. This is not an easy task. It is widely known that there is an important gap between BPs and rules indeed [10]. Different compliance rule modelling languages and ways to insert policy-related information in BP models have been introduced in the last years [11,3,12,13], but there are important problems that still remain partially unsolved. The own nature of rules is a problem itself, as policies are described in natural language, which may be ambiguous. Therefore, two organizations may implement the same rule in two slightly different ways, sometimes voluntarily (i.e. for "business policy"), other times by chance (i.e. due to an unconscious different interpretation or a misunderstanding of the rule). For instance, let us suppose we must fulfill the following rule in our organization:

Rule 1: *There is a segregation of duties between the creation of a hiring resolution proposal and its processing.*

Segregation of duties (SoD) is a well-known authorization constraint that aims at avoiding problems due to a conflict of interests in the execution of two activities. This is achieved by distributing the responsibility to more than one resource (e.g. roles, positions, persons). If we are developing a rule modelling language, the SoD concept may be itself an element of the language that can be used to state rules such as Rule 1. However, by acting like that we are losing nuances that come from the human interpretation of natural language. For example, Rule 1 could lead to at least two different implementations:

– Strict. The business manager can assume that, besides selecting different roles for the two tasks, it is necessary to guarantee that two different persons undertake the activities in order to prevent the scenario in which the same person plays the two roles involved and, hence, may execute the two activities affected by the rule, which could result in a conflict of interests.
– Slight. However, we may find an organization in which people must indicate with which role perform every task and which does not care about the same person executing the two activities involved in a SoD as long as each activity is performed with a different role. In this case, the result of the SoD checking would be different from that of the previous implementation.

Each piece of our rule modelling language must have an associated semantics. So, the question is: which of the previous implementations is covered by our language? Both? Taking into account the whole potential interpretation variability of natural language would lead to complex languages, which in turn would derive in their hard understandability and use.

Another issue is how to deal with all the aspects of the BPs that can be affected by rules (e.g. control flow, data, resources, temporal constraints). The wide bunch of possibilities regarding BP aspects collaborate in making it difficult to develop such a language. Furthermore, some rules involve more than one aspect. This, together with the aforementioned interpretation problem, generate a complex and varied casuistry that makes it hard to define a *declarative* domain-specific language (DSL) expressive enough to address all kinds of rules. As a consequence, most of the techniques proposed so far limit their scope, e.g. to one or two BP aspects, giving rise to many ad-hoc approaches. The conclusions of a study we carried out on approaches dealing with BP compliance can be found in [14]. Besides, some techniques rely on one specification formalism for rule definition such as [13]. However, one formalism usually allows only some types of checks, so the entire casuistry is not covered like that. The mixture of different formalisms would be necessary.

Finally, parts of some rules can be re-used in the definition of other rules. Considering this in our rule modelling language would save effort to business analysts or to the person in charge of modelling the rules of a company (e.g. a compliance expert). Some pattern-based approaches such as BPMN-Q [12] kind of cover this issue. However, the problem of the large casuistry is still latent.

We propose *mashups* as a mechanism to provide an *operative* specification of rules and check design-time BP compliance. In this paper we will reveal how this approach allows us to address the aforementioned challenges and overcome the aforementioned issues. Mashups are easy to understand and use [15], and they can be implemented in many different ways, e.g. by using common spreadsheets [16]. Applied to policies, mashups let us handle different interpretations of natural language by re-combining the mashup components used to specify a rule, thus dealing with natural language ambiguity.

Furthermore, different formalisms can be used together in a single mashup, provided that we have a real way to connect the information resulting from a technique with the input of another approach[1]. Therefore, the specification of rules by an end user may be independent of the underlying formalism for compliance checking. Besides, mashups offer, among other interesting features, flexibility, portability and re-usability, so rules already defined (or part of them) can be saved and used later to build other mashups. The fact that they have already been used for analysis purposes in other domains [17,18], motivated us to explore their applicability to check BP compliance rules.

Finally, note that, although the focus of this paper is on DTCC, the idea behind our approach can be applied to other phases of the BP lifecycle and, as a matter of fact, we are currently working on extending the approach to RTCC and BCC.

The rest of the paper is structured as follows. In Section 2 we introduce mashups and their main types of components, accompanied by a generic example. Section 3 contains an explanation of our proposal, which is exemplified by applying it to a specific kind of compliance rules in Section 4. Finally, some conclusions and a summary of ongoing and future work are presented in Section 5. To improve the understandability of the explanations, and due to space limitation, references to related work are given throughout the paper.

2 Introduction to Mashups

Mashups are a hot concept in IT nowadays. A mashup is a data-driven workflow (a.k.a. *dataflow*) built with information from one or more data sources, and it is based on the re-use of contents and functionalities. Mashups were developed to build new Web services or applications from existing data in an "easy" way, so the end user does not need to have specific technical knowledge, but only knowledge of the problem domain [15]. They have already been applied to address problems such as the analysis of molecular biology in bioinformatics [17] and the simplification of patient management in hospitals [18], and there are some mashup makers in the market such as Intel Mashup Maker, Yahoo! Pipes or IBM Mashup Center. The Google App Engine also gives support to the previous Google Mashup Editor. There is a large amount of mashup examples available on the Web, e.g. more than 6,000 mashups can be found at `http://www.programmableweb.com/`. With tools such as Yahoo! Pipes anybody can build and publish a mashup in the Internet.

To show mashup appearance and use we have created the mashup in Figure 1. It returns the last 25 international news of the New York Times and The Australian digital newspapers[2]. Any researcher could want to have a similar

[1] The complexity of a mashup is within each component, and the greatest effort is put on how to integrate them.

[2] It can be run at `http://pipes.yahoo.com/cabanillasmashups/worldnews`.

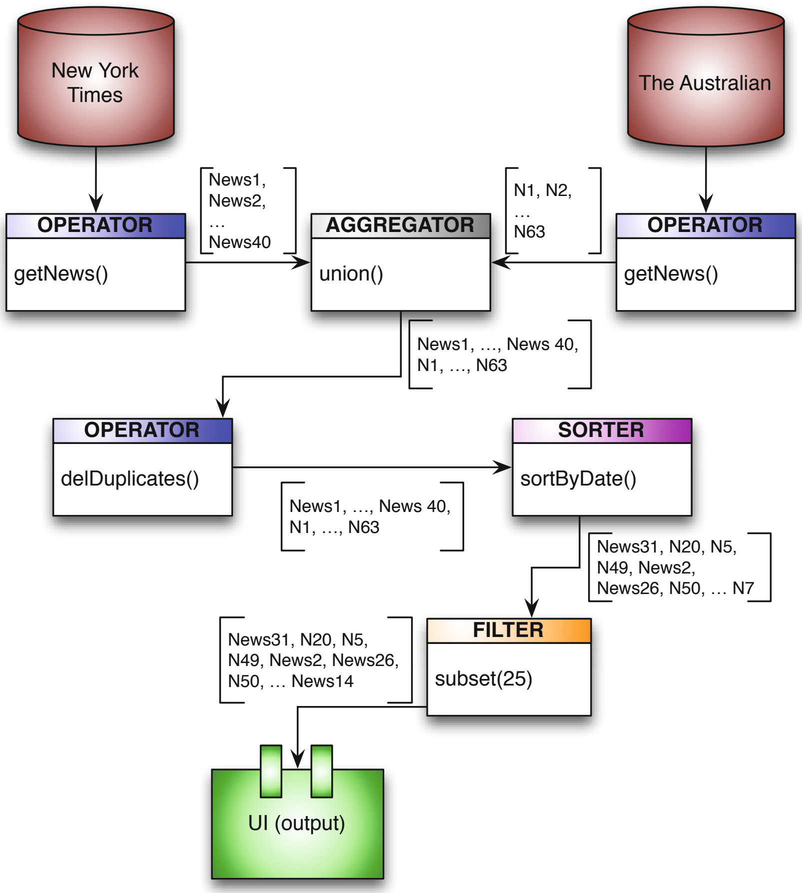


Fig. 1. Mashup collecting the last 25 world news from two digital newspapers

mashup to be kept up to date of his/her research interests, e.g. a mashup that automatically places on a map the venue cities where the next editions of his/her favourite conferences take place.

As illustrated in the figure, mashup editors allow the definition of the dataflow by connecting two main types of components: data sources and flow components.

Data sources range from data warehouses to URLs pointing at RSS feeds or any kind of accessible information.

Flow components are the elements in charge of operating on data, so they all have input and output. The input data they receive can come from another mashup component or from a data source, and the last component provides the information required to the output UI. Flow components can be generic-purpose components such as those that handle collections of elements to sort or join them, and domain-specific (DS) components, which implement functions specific to the problem domain, such as handling geographical location data to enrich Google Maps with external information. Some frequently used flow components include the following:

- *Filters.* They narrow down the flow of data, supporting the transformation of the information.
- *Aggregators.* They join or group data according to some criteria.
- *Operators.* They extract, elaborate and transform the information, constituting a very important part of the ETL (Extract-Transform-Load) process [19] data must undergo from the input of the mashup to the output UI. They range from operators that implement well-known functions such as *count*, *min*, *max* and *average* (i.e. general-purpose) to operators that handle strings or extract and builds geographical location information (i.e. domain-specific).
- *Sorters.* They return the same input data but in a specific order.

Languages for mashup representation, such as Enterprise Mashup Markup Language (EMML)[3], provide support to create and use at least the aforementioned components and they can usually be extended to include new DS components, if required. For insights about how to build high-quality mashups we recommend the reading of [20].

3 Mashups for BP Compliance Checking

We propose the use of mashups as *a language to provide an operative specification of rules and to query BP models.*

Definition 1. A *compliance mashup* is an operative DSL that allows the integration of heterogeneous data sources and the specification of compliance rules over subsets of the information that can be extracted from them.

In them, the data sources are the repositories where the organization stores the different models it uses, e.g. business processes, organizational structures, data and so on. Regarding the flow components of the mashup specification, a set of both general-purpose and DS components (filters, operators, sorters, etcetera) may be necessary to manipulate and transform the data coming from these models. In particular, in the case of DS operators, these components will encapsulate *analysis operations* (or queries) on models that enable the creator of the mashup to extract from the models the information he/she needs to check a compliance rule.

DS operators can be implemented using different analysis techniques. For instance, BPMN-Q is a language aimed at querying BP models regarding control flow and data [21] (e.g. it returns information on whether an activity is executed before or after another activity). There also exist mechanisms to analyse the data perspective of BP models, as long as these models have data-related information [22]. Other proposals deal with resource analysis in BP models with resource assignment information such as RAL or Business Activities. RAL is a DSL for the representation and analysis of resource assignments in BP models. The analysis

[3] http://www.openmashup.org/omadocs/v1.0/index.html

mechanism is provided by means of its ontology-based semantics [23]. Business Activities are a UML extension to integrate process flows and process-related RBAC models with resource-related constraints. The violation of constraints such as mutual exclusion between activities can be detected [24].

The set of available DS operators will depend on the kinds of checks we have to perform over the BPs of our organization. So, as aforementioned, some of them will implement mechanisms to check for some data-related functionality, some others will be focused on dealing with control-flow information, and so on.

Finally, the data sources and the flow components are connected in a dataflow to check for compliance rules.

As can be seen, building a mashup is not a hard task, assuming that all the logic within the components is implemented. They thus allow us to re-use existing solutions, avoiding to re-invent the wheel.

4 Applying Mashups to Resource-Related Compliance by Design

We are going to show how to build a mashup to specify and check rules related to resource management in BPs at design time. It is one of the aspects usually affected by rules nowadays, as we can find plenty of policies that regulate resource management in a company, to be named:

– SoD is well-known in financial accounting systems as a mechanism to prevent from fraud and error. In IS, it helps reduce the potential damage from one person's actions by disseminating the tasks and associated privileges for a specific BP among multiple users. A big part of the Sarbanes-Oxley Act (SOX)[4] is devoted to manage internal control in IT, in which SoD is a key concept.
– The Health Insurance Portability and Accountability Act (HIPAA)[5] pays special attention to who can do certain tasks in order to preserve the privacy of confidential information and to avoid fraudulent use of private data.
– Besides rules coming from well-known policies, each company has its own resource-related business rules that are defined ad-hoc to its BPs, e.g. to state what kind of resource (e.g. a role or a specific person) is in charge of each task.

As a use case we will use a real process designed and utilised in the Andalusian Institute of Public Administration (IAAP) that represents the procedure to create and process a resource resolution proposal for hiring people. This process has a high use frequency in the Andalusian Public Administration, which serves to more than 8 million end users. It has been modelled in BPMN for the ease of understanding (cf. Figure 2). As depicted in the model, once a draft of a resource resolution proposal is created, it is concurrently sent to the Consultative

[4] http://www.soxlaw.com/
[5] http://www.hhs.gov/ocr/privacy/

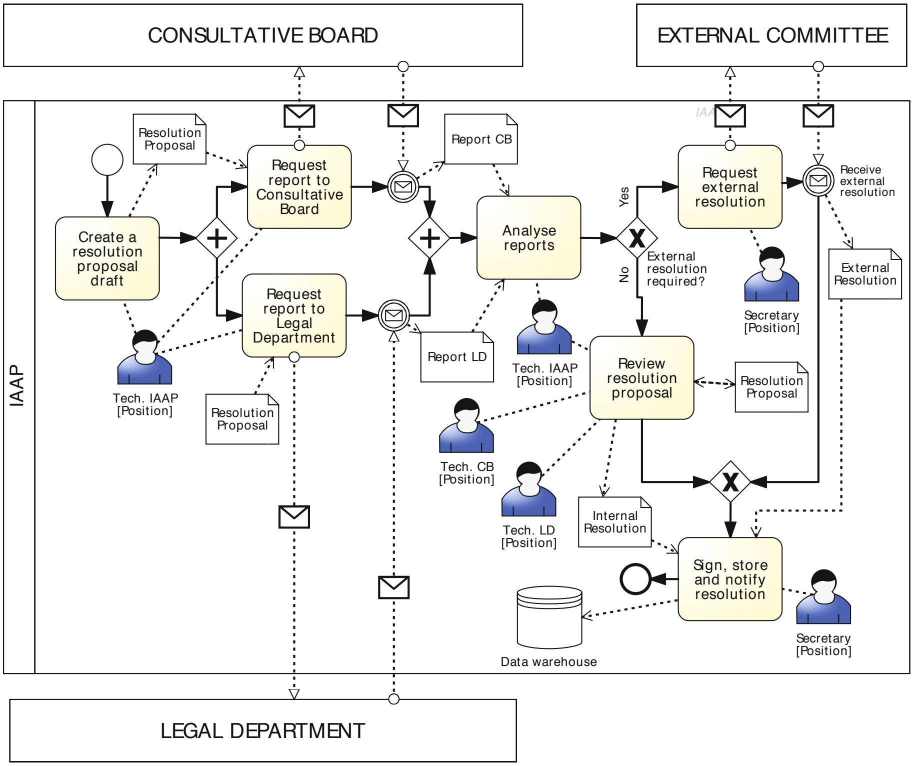
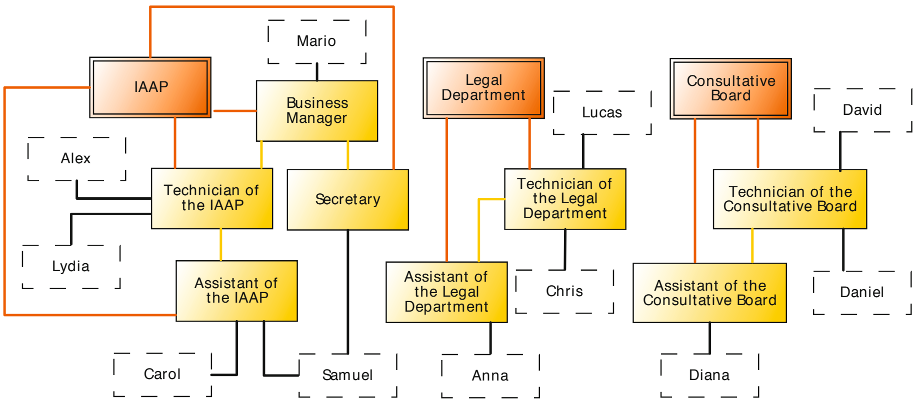


Fig. 2. Excerpt of the process to create and process a resource resolution proposal

Board and to the Legal Department for it to be evaluated. After receiving both evaluations the IAAP analyses them and decides whether an external resolution is required. In that case, a request is sent to an external committee, which must create and send a new resolution. Otherwise, the resolution proposal is reviewed and changes are applied to the initial draft. In any case, the documents generated are signed and archived, and the resolution result is appropriately notified.

Since we focus on resource-related compliance checking, we must be aware of the organizational structure of the IAAP. Figure 3 shows it regarding Administrative Resource Management. There are three organizational units called IAAP, Legal Department and Consultative Board, corresponding to different work teams. Eight positions (Business Manager, Technician of the IAAP, Assistant of the IAAP, Secretary, Assistant of the Legal Department, Technician of the Legal Department, Assistant of the Consultative Board, and Technician of the Consultative Board), occupied by a total of eleven people[6] (shown in white dash-lined rectangles in the figure), are associated to these units. Positions are connected to each other to represent hierarchical relations between them.

[6] Their names have been changed for privacy reasons.



Position	Role
Business Manager	Accountable
Technician of the IAAP	Technician Analyst
Technician of the Legal Department	
Technician of the Consultative Board	
Assistant of the IAAP	Technical assistant Document writer
Assistant of the Legal Department	
Assistant of the Consultative Board	
Secretary	Document signer

Fig. 3. Excerpt of the organizational structure of the IAAP

Figure 2 also depicts the resource assignments to the process activities. Note that assigning several resources to an activity means any of them can execute it, e.g. activity *Review resolution proposal* can be done by a Technician of the IAAP, a Technician of the Consultative Board or a Technician of the Legal Department.

Given this scenario we are interested in specifying and checking Rule 2:

Rule 2: *The creation of the resolution proposal draft and its revision after the evaluations of the Consultative Board and the Legal Department have to be performed by different roles.*

The Business Manager of the IAAP would be very interested in knowing if Rule 2 is met given the current BP model and the organizational structure in order to do the proper changes before running the process, if necessary. As we are checking the rule at design time, some considerations have to be done. Specifically, we should check that given the current resource assignments to activities, the same role can *never* execute the two mutual exclusive activities. Otherwise we can consider the process as non-compliant because we cannot ensure the rule is always fulfilled, that is, it could be met or violated depending on the specific resource allocation carried out at run time.

Figure 4 defines Rule 2 following the aforementioned consideration. In this compliance mashup we have two different data sources: (i) a repository of resource-aware BP models, i.e. models enriched with resource assignments; and

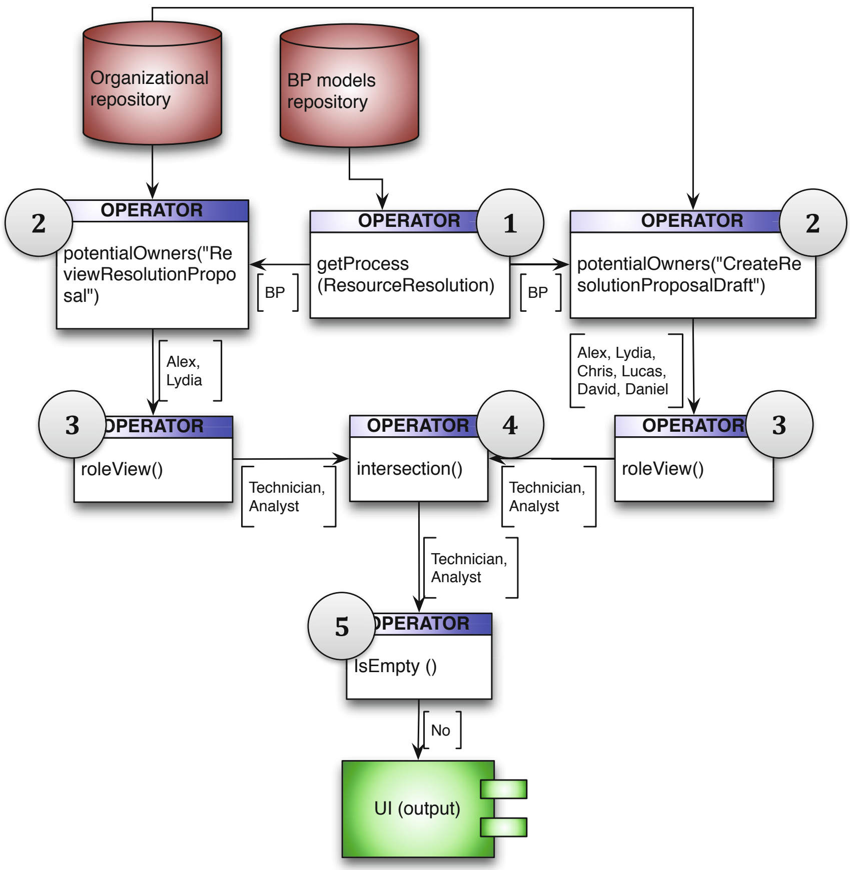


Fig. 4. Compliance mashup to define and check Rule 2

(ii) a repository of information related to the resources of the company (i.e. its organizational model, information about each resource and the like). The steps given to specify and check the rule are the following:

1. The BP affected by the rule is extracted from the repository of BP models.
2. The potential performers/owners (i.e., who *may* be allocated an activity at run time) of the two activities involved in Rule 2 are calculated.
3. As we are interested in roles, we generate an output on the basis of potential roles instead of potential persons, for which information from the organizational model is required (cf. Figure 3 to understand better).
4. Only the roles common to both sets of potential performers/owners must be selected.
5. If the resulting set is empty we can assure the rule is being always met. Otherwise we should state the rule may be violated at run time.

Therefore, in this concrete case the BP model is not compliant with Rule 2.

As shown, the mashup is easily built by selecting and connecting the proper components. Their inner implementation is not important to end users. Therefore, non-technical people from our organization could benefit from several mashup platforms that let any person with knowledge on the application domain create a mashup, with no need of technical knowledge. The functionality of the mashup could even be embedded in other existing applications, e.g. Kongdenfha et al. [16] have presented an approach that allows users to easily build mashups within a familiar spreadsheet environment.

If we wanted to give a slightly different meaning (or materialization) to Rule 2, we could re-configure the properties of the convenient components in Figure 4, reconnect them, or even insert new components to deal with the new interpretation (e.g. to force different people performing the two activities involved).

Besides, mashups can return different types of results (e.g. numeric values, text, boolean values). It means the output UI can show not only compliance checking results, but also the result of any *analysis operation* over BP models. For instance, we could build a mashup to obtain the list of activities that can be executed after a given activity. Therefore, we can use mashups for different kinds of analyses related to BP compliance and, in general, to BP management, always relying on the basis of re-using and integrating techniques. New approaches could be implemented and included as new components.

4.1 DS Components for Resource-Related Compliance Mashups

Most of the flow components depicted in Figure 4 are DS elements, specifically DS operators. Table 1 depicts some DS operators in the domain of resource-related compliance checking. Implementing some of them is quite trivial, e.g. those focused on the extraction of specific information from BP models or organizational models, as long as we have proper mechanisms to access the data. However, there are other DS operators whose implementation may not be so easy. That is the case of *potentialOwners*, in which we have to figure out what the candidates to execute the task at run time are from the resource assignments in the BP model. This involves an *analysis* of the BP model and the organizational model together to obtain the required information. Several approaches have been developed so far to perform this kind of analysis. For instance, RAL Solver, based on the aforementioned RAL, uses the HermiT DL reasoner to resolve RAL expressions and implement all of the operations depicted in Table 1. Similarly, many of those operations can also be implemented using the aforementioned Business Activities [25], or by means of the model-checking algorithms proposed by Wolter et al. to verify control access constraints in BPs [26]. Any of these approaches (or others) could be used to execute the mashup in Figure 4.

It is important to point out that the list of DS operators presented in Table 1 is not exhaustive. Indeed, operations to check properties related to dynamic issues in BPs are also necessary in order to consider the control flow of the process. Formalisms such as Petri nets [27] and BPMN-Q [12] have been widely studied and used in this kind of checks. Flow components performing functions for control-flow management could thus be incorporated to the mashup.

Table 1. DS operators for resource-related compliance checking

DS operation	Output	Function
Operations on business process models		
getProcess (PID)	BP	It returns the BP elements of the selected BP model
getActivities (PID)	[Activity]	It returns the set of activities of the selected BP model
Operations on organizational models		
getPerson (ID)	Person	They return the information associated to a specific element of an organizational model
getPosition (ID)	Position	
getRole(ID)	Role	
getUnit(ID)	Unit	
Operations using information from several data sources		
potentialOwners (A, OM)	[Person]	Potential performers of an activity according to its associated resource assignment expression. Sometimes looking through the organizational model may be required
potentialOwners ([Activity], OM)	[Activity: [Person]]	Potential performers of a set of activities. Sometimes looking through the organizational model may be required
positionView ([Activity: [Person]], OM)	[Activity: [Position]]	It classifies the persons by positions (according to the organization structure) and shows the positions associated with the activities
roleView ([Activity: [Person]], OM)	[Activity: [Role]]	This operation classifies the persons by roles (according to the organization structure) and shows the roles associated with the activities
unitView ([Activity: [Person]], OM)	[Activity: [Unit]]	It classifies the persons by organizational units (according to the organization structure) and shows the units associated with the activities

5 Conclusions and Future Work

The use of mashups to specify and check compliance rules provides a framework that may help to simplify an important problem in business process compliance management, which is the expression of compliance rules in a way that enables their automated checking. Since rules are defined by composing operators, mashups provide flexibility to give stricter or slighter meanings to them, as desired. Besides, mashups are quite easily understandable, which allows their creation or modification by people without technical skills. Furthermore, compositions can be saved as new mashup components, making them reusable.

This proposal provides a common approach to specify and check compliance rules regardless of the compliance domain provided as long as there is technical and technological support to implement the components for the specific domain. That is, we have applied mashups in DTCC, but with different data sources and DS flow components we could check other kinds of compliance. For example, building mashups for BCC would give rise to UIs that could work as a dashboard for the visualization of compliance issues for later audits [28].

Regarding the validation of our approach, we have just started a project with an IBEX-35 multinational company. The goal of the project is to develop a mashup-based system that allows the specification and checking of all the rules the company has to meet (basically rules from Internal Control Systems of Financial Information –SCIIF–, law decree L262, SOX, and some ad-hoc business rules). In the policies the company needs to check we have already found cases of ambiguous statement of rules in which a specific interpretation must be given, e.g. "database security parameters must be checked at least once a year". The following question emerged when modelling this rule: "a fiscal year or a year since the last time it was checked?" This reaffirms the need of providing flexibility for rule modelling.

Besides, the solution must cover all the compliance-related features described in [9], giving rise to a full-coverage BPCMS. Therefore, we are studying how to extend the approach presented in this paper to address more types of BP compliance (e.g. run-time issues). In this project, we also plan to evaluate how easy the building of compliance mashups actually is for non-technical people.

References

1. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer (2007)
2. Kharbili, M.E., de Medeiros, A.K.A., Stein, S., van der Aalst, W.M.P.: Business process compliance checking: Current state and future challenges. In: MobIS 2008, pp. 107–113 (2008)
3. Liu, Y., Muller, S., Xu, K.: A static compliance-checking framework for business process models. IBM Systems Journal 46, 335–362 (2007)
4. Ghose, A.K., Koliadis, G.: Auditing Business Process Compliance. In: Krämer, B.J., Lin, K.-J., Narasimhan, P. (eds.) ICSOC 2007. LNCS, vol. 4749, pp. 169–180. Springer, Heidelberg (2007)
5. Weber, I., Governatori, G., Hoffmann, J.: Approximate Compliance Checking for Annotated Process Models. In: Advances in Enterprise Engineering - GRCIS workshop at CAiSE 2008 (June 2008)
6. Silveira, P., Rodríguez, C., Birukou, A., Casati, F., Daniel, F., D'Andrea, V., Worledge, C., Taheri, Z.: Aiding compliance governance in service-based business processes. In: Non-Functional Properties for Service-Oriented Systems: Future Directions (NFPSLA-BOOK-2011). IGI Global (2011)
7. van der Aalst, W.M.P., van Dongen, B.F., Günther, C.W., Rozinat, A., Verbeek, E., Weijters, T.: ProM: The Process Mining Toolkit. BPM (Demos) (2009)
8. Rozinat, A., van der Aalst, W.M.P.: Conformance checking of processes based on monitoring real behavior. Information Systems 33(1), 64–95 (2008)
9. Cabanillas, C., Resinas, M., Ruiz-Cortés, A.: Exploring Features of a Full-Coverage Integrated Solution for Business Process Compliance. In: Salinesi, C., Pastor, O. (eds.) CAiSE Workshops 2011. LNBIP, vol. 83, pp. 218–227. Springer, Heidelberg (2011)
10. Awad, A., Weidlich, M., Weske, M.: Visually specifying compliance rules and explaining their violations for business processes. J. Vis. Lang. Comput. 22, 30–55 (2011)
11. Yu, J., Manh, T.P., Han, J., Jin, Y., Han, Y., Wang, J.: Pattern Based Property Specification and Verification for Service Composition. In: Aberer, K., Peng, Z., Rundensteiner, E.A., Zhang, Y., Li, X. (eds.) WISE 2006. LNCS, vol. 4255, pp. 156–168. Springer, Heidelberg (2006)

12. Awad, A., Decker, G., Weske, M.: Efficient Compliance Checking Using BPMN-Q and Temporal Logic. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) BPM 2008. LNCS, vol. 5240, pp. 326–341. Springer, Heidelberg (2008)
13. Governatori, G., Sadiq, S.: The Journey to Business Process Compliance. In: Handbook of Research on BPM, pp. 426–454. IGI Global (2009)
14. Cabanillas, C., Resinas, M., Ruiz-Cortés, A.: Hints on how to face business process compliance. In: III Taller de Procesos de Negocio e Ingeniería de Servicios (PNIS 2010) en JISBD 2010, pp. 26–32 (2010)
15. Yu, J., Benatallah, B., Casati, F., Daniel, F.: Understanding Mashup Development. IEEE Internet Computing 12, 44–52 (2008)
16. Kongdenfha, W., Benatallah, B., Vayssière, J., Saint-Paul, R., Casati, F.: Rapid development of spreadsheet-based web mashups. In: WWW 2009, pp. 851–860 (2009)
17. Hull, D., Wolstencroft, K., Stevens, R., Goble, C.A., Pocock, M.R., Li, P., Oinn, T.: Taverna: a tool for building and running workflows of services. Nucleic Acids Research 34(Web-Server-Issue), 729–732 (2006)
18. Baresi, L., Guinea, S.: Mashups with Mashlight. In: Maglio, P.P., Weske, M., Yang, J., Fantinato, M. (eds.) ICSOC 2010. LNCS, vol. 6470, pp. 711–712. Springer, Heidelberg (2010)
19. Akkaoui, Z.E., Zimanyi, E.: Defining ETL worfklows using BPMN and BPEL. In: Proceeding of the ACM Twelfth International Workshop on Data Warehousing and OLAP, Hong Kong, China, pp. 41–48. ACM (2009)
20. Cappiello, C., Daniel, F., Koschmider, A., Matera, M., Picozzi, M.: A Quality Model for Mashups. In: Auer, S., Díaz, O., Papadopoulos, G.A. (eds.) ICWE 2011. LNCS, vol. 6757, pp. 137–151. Springer, Heidelberg (2011)
21. Sakr, S., Awad, A.: A framework for querying graph-based business process models. In: WWW 2010, pp. 1297–1300 (2010)
22. Cabanillas, C., Resinas, M., Ruiz-Cortés, A., Awad, A.: Automatic Generation of a Data-Centered view of Business Processes. In: Mouratidis, H., Rolland, C. (eds.) CAiSE 2011. LNCS, vol. 6741, pp. 352–366. Springer, Heidelberg (2011)
23. Cabanillas, C., Resinas, M., Ruiz-Cortés, A.: Defining and Analysing Resource Assignments in Business Processes with RAL. In: Kappel, G., Maamar, Z., Motahari-Nezhad, H.R. (eds.) ICSOC 2011. LNCS, vol. 7084, pp. 477–486. Springer, Heidelberg (2011)
24. Strembeck, M., Mendling, J.: Modeling process-related RBAC models with extended UML activity models. Inf. Softw. Technol. 53, 456–483 (2011)
25. Schefer, S., Strembeck, M., Mendling, J., Baumgrass, A.: Detecting and Resolving Conflicts of Mutual-Exclusion and Binding Constraints in a Business Process Context. In: Meersman, R., Dillon, T., Herrero, P., Kumar, A., Reichert, M., Qing, L., Ooi, B.-C., Damiani, E., Schmidt, D.C., White, J., Hauswirth, M., Hitzler, P., Mohania, M. (eds.) OTM 2011, Part I. LNCS, vol. 7044, pp. 329–346. Springer, Heidelberg (2011)
26. Wolter, C., Miseldine, P., Meinel, C.: Verification of Business Process Entailment Constraints Using SPIN. In: Massacci, F., Redwine Jr., S.T., Zannone, N. (eds.) ESSoS 2009. LNCS, vol. 5429, pp. 1–15. Springer, Heidelberg (2009)
27. van der Aalst, W.M.P.: The Application of Petri Nets to Workflow Management. Journal of Circuits, Systems, and Computers 8(1), 21–66 (1998)
28. Silveira, P., Rodriguez, C., Casati, F., Daniel, F., D'Andrea, V., Worledge, C., Taheri, Z.: On the Design of Compliance Governance Dashboards for Effective Compliance and Audit Management. In: NFPSLAM-SOC 2009 (2009)

Integrity of Supply Chain Visibility: Linking Information to the Physical World

Joris Hulstijn[1], Sietse Overbeek[1], Huib Aldewereld[1], and Rob Christiaanse[2]

[1] Faculty of Technology, Policy and Management, Delft University of Technology, Jaffalaan 5, 2628 BX Delft, The Netherlands
{J.Hulstijn,S.J.Overbeek,H.M.Aldewereld}@tudelft.nl,
[2] VU University Amsterdam and EFCO Solutions, Singel 318, 1016 AE Amsterdam
r.christiaanse@efco-solutions.nl

Abstract. Regulatory compliance in international trade can be enhanced by facilitating electronic exchange of trade documents to increase the supply chain visibility. Crucial for acceptance of the supply chain visibility concept is trust in the reliability of the data. This depends on both the integrity of information (no data is altered illicitly) and integrity of the flow of goods (no goods are unknowingly added or taken away). The challenge is to determine how these concepts of integrity are interconnected. In this paper, we discuss control measures to ensure integrity of supply chain information and the related goods flow. Such controls consist of three components. First, only trusted traders take part. Second, technical control measures ensure internal consistency of the information system, i.e., the system maintains integrity constraints based on a model of the transactions taking place. Third, physical and organizational control measures mitigate the risk that events in the real world do not correspond to the reported transactions. The usefulness and adequacy of the approach is illustrated with two case studies: the use of electronic seals in the Smart-CM project, and the use of biometrics for authentication in the E-Link project at Schiphol Airport, the Netherlands.

Keywords: integrity, supply chain visibility, virtual data pipeline.

1 Introduction

Customs organizations are redefining their role regarding dealing with goods, not persons. On the one hand, they are supposed to facilitate international trade and reduce the related accounting burden for businesses. On the other hand, they must enhance safety and security of the international supply chain, in addition to their fiscal tasks. Generally, customs use a risk-based approach for selecting freights for inspection. These risk assessments require reliable data about transportation movements and the contents of containers. However, currently, the supply chain is organized in such a way that such data is not always reliable [13]. For instance, the *customs manifesto*, which is the declaration that must be sent as an advance warning of importing goods is often incomplete or incorrect.

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 351–365, 2012.

One way to solve this problem is to enhance *supply chain visibility*: information systems help to trace evidence of the flow of goods along the supply chain [14]. When trading partners and regulators cooperate, it becomes possible to reuse commercially available information for regulatory purposes. This is called *piggy backing* [21]. Commercially available information tends to be of better quality than information that is only demanded for regulatory reasons, because trading parties typically have countervailing interests. They will verify each others' data to see whether documents correspond to the actual flow of money and goods.

A particular approach to enhance supply chain visibility is the *virtual data pipeline*, developed in the CASSANDRA research project [14]. The virtual data pipeline is essentially a data sharing architecture based on Linked Open Data [25]. Linked Open Data refers to data published on the Web in such a way that it is machine readable, its meaning is explicitly defined, it is linked to other external datasets, and it can in turn be linked to from external datasets as well. Linked Open Data also refers to a set of best practices for publishing and connecting structured data. Here it will be used for exchanging trade documents among trusted traders. The regulator can view regulatory documents, e.g., customs declarations as well as selected commercial trade documents.

Figure 1 shows the virtual pipeline with the physical flow of goods underneath. The process follows the various stages of packing and sending the goods, freight forwarding, transportation, warehousing and distribution, until finally containers are unpacked by the consignee. The consigner is another word for seller of the goods and the consignee is the buyer.

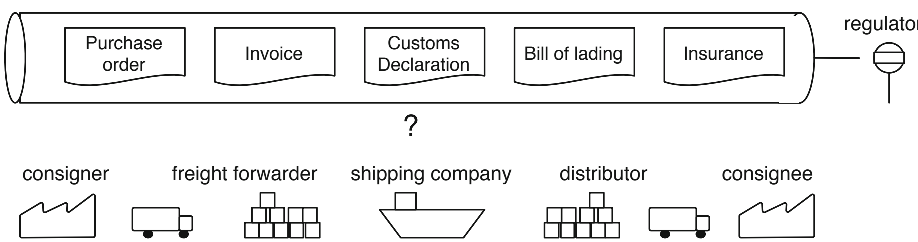


Fig. 1. Integrity of the virtual data pipeline concept

Crucial for acceptance of the data pipeline concept is the reliability of the data being exchanged. In this paper, we focus on the related notion of integrity. Literally integrity means 'undivided', 'whole' or 'without flaws'. The notion can be applied to the physical flow of goods. In that case it means that no goods may be added or taken from the flow, without this being noticed. Applied to data, integrity means: "Guarding against improper information modification or destruction, and includes ensuring information non-repudiation and authenticity" [8]. Information integrity involves accuracy (corresponds to reality) and completeness (covers all relevant aspects of reality), as well as timeliness (up-to-date) and validity (collected and processed according to regulations) [3]. Therefore, establishing integrity requires an audit trail, a record of all steps in the process, which makes it possible to re-generate and verify information when needed.

In the context of supply chain visibility, the two notions of integrity should be interconnected, i.e., the information integrity and the integrity of the flow of goods. The central research question therefore is:

> How can we make sure that the data being exchanged in the virtual data pipeline corresponds to all relevant aspects of the physical flow of goods?

The contribution of our approach is to define a business process model as a kind of intermediate stage. Steps in the process constitute a status update, which, when taken together form a well-defined economic transaction. Each status update is represented by a message or trade document, which is made available through the virtual data pipeline. Moreover, each status update is associated with an event or activity in the physical flow of goods of which evidence is also recorded. To undeniably link the different kinds of evidence, we need at least three kinds of control measures: (i) certification to ensure only known and trusted traders take part, (ii) control measures to ensure internal consistency, i.e., all data adheres to integrity constraints based on well-formed transactions [5], and (iii) control measures to ensure external consistency, i.e., to ensure that data will undeniably correspond to reality [5]. Our working hypothesis is that these three kinds of measures are necessary and sufficient, to establish integrity of the data about the flow of goods. A more general claim is that an intermediate conceptual model stated in terms of status updates, will in fact facilitate the design of control measures to ensure integrity of data about the flow of goods. The idea is that integrity can be established by aligning such status updates.

The approach is illustrated by two cases: (1) the E-Link project at Schiphol Airport, for redesigning customs procedures for air cargo export, and (2) the use of electronic seals for securing containers, as developed in the Smart-CM project. We also discuss challenges to the feasibility of the virtual data pipeline concept.

The remainder of the paper is structured as follows. In Section 2 we present our approach for ensuring integrity of the data about the flow of goods. Section 3 contains the two cases. Section 4 contains a discussion about feasibility. The paper ends with conclusions and recommendations for further research in section 5.

2 Linking Information Integrity and Physical Integrity

The solution can be modeled in terms of a three layer architecture (Figure 2), compare Dietz [7]. On the bottom we have the physical layer with activities and events that constitute the transportation and distribution of goods. On the top we have the value layer with a business process which constitutes an economic transaction: generally a transfer of ownership or custody over the goods in return for money. The layer in the middle is the information layer. In this layer, evidence of events at the other two layers is being recorded and processed.

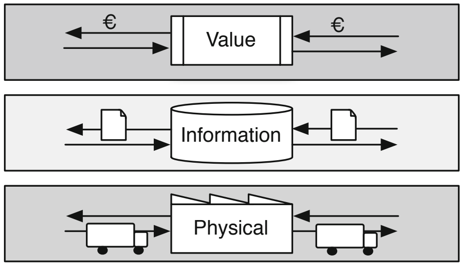


Fig. 2. Linking physical events through information exchange to value transactions

Generally, each relevant activity in the physical flow of goods is associated with a step in the business process. For example, when goods are unloaded from a truck, custody over the goods is transferred. Moreover, each status update is associated with a message or trade document, recorded in the participants' information systems. For example, when goods are unloaded the inventory master will sign a receipt for the goods, which is called a quittance. This piece of evidence will acquit the truck driver from the responsibility for the goods. Our purpose here is to reengineer the system of transfers and quittances in such way, that all major status changes in the business process are accounted for by undeniable evidence linked to physical events.

Each layer displays a kind of storage. On the physical layer a container, warehouse or ship contains inventory. In the middle we find a database. On the top, we find an abstract value repository, with financial accounts representing commitments and assets. At each layer, activities and events constitute status updates, which may add or take things away. Each storage concept can therefore be seen as a kind of data structure, subject to an *invariant*: a set of constraints which must be true at all times. Events may temporarily invalidate an invariant, but after a transaction the invariant will be restored. This is what constitutes a well-formed transaction: it restores the invariant, see Section 2.5 below.

For each storage concept, there are general *continuity equations* [22], compare reconciliation relations in accounting [20]. For example, inventory at the end of the day should equal inventory at the beginning, with all incoming goods added and all outgoing goods subtracted. Or, the sum total of incoming goods of party x should equal the outgoing goods of upstream partner $x-1$. Also for information similar equations can be formulated. For instance, the state of a database at the end of the day, should equal that of applying all updates to the begin state.

$$end = begin + in - out \tag{1}$$

$$in(x) = out(x-1) \tag{2}$$

The vertical relationship between layers is representational: events at the information layer represent status changes, both at the value layer above and the physical layer below. Conversely, the semantics of the concepts at the information layer can be expressed in terms of the value concepts or physical concepts.

In other words: they provide evidence of status changes. The notion of evidence is based on so-called constitutive rules or counts-as rules [19]. Representations are social conventions, which only make sense within a certain social or institutional context. Therefore, it is important that the virtual data pipeline is a closed environment in which all participants are known.

2.1 Trusted Traders

The virtual data pipeline is essentially a closed system: participants are known and should in principle be trustworthy. In the context of the CASSANDRA project, traders are trusted when they are certified by the customs authority. They are so-called Authorized Economic Operators (AEO) [6]. In particular, the procedure uses a self-assessment which involves a risk analysis. After that, customs officials make an assessment whether the proposed control measures are sufficient given the risks on the basis of: (1) previous compliance behavior towards customs, (2) financial solvability, (3) reliable administration, and (4) adequate safety measures given the risks of the business.

In addition to certification by the authorities, we expect that commercial partners may also require a vetting procedure. After all, in general companies are reluctant to share information in a supply chain [16]. Participants may collaborate in one instance but be competitors in another instance. Note that a supply chain is flexible; the network infrastructure is stable. Therefore, the system should maintain who is collaborating with whom on a contract basis.

2.2 Evidence and 'Counts as'

In all complex social systems, events or situations in reality (*brute facts*) are given an additional social or institutional meaning (*institutional facts*). A quittance is an example. The brute fact is the piece of paper that one receives upon delivery of goods, while institutionally it is seen as a waiver of responsibility. This layering of institutional meaning upon real world events is based on John Searle's notion of 'constructing social reality' [19]. Social reality is created through *counts-as* statements: in institution S, fact or event X *counts as* fact or event Y.

Table 1. Three notions of counts-as, producing evidence of a waiver of responsibility

1.	"In institution S, a receipt created by procedure P *counts-as* a quittance."	**Constitutive**
2.	"A quittance *counts-as* a waiver of responsibility."	**Classificatory**
3.	"In institution S a receipt created by procedure P *counts-as* a waiver of responsibility."	**Proper Classification**

The concept of counts-as has been thoroughly studied in formal logics [12], but also in computer science [1]. Grossi [12] first identified that the notion of counts-as has three separate readings, see Table 1. The second premise is a general classification rule. It is universally applicable and could thus be formalized as ontological subsumption or logical implication, as in [1]. The conclusion, in 3, is a classification which only holds with respect to a given context. The first premise is not classificatory, but is 'constitutive': it can change institutional reality.

This formal link between brute facts and institutional facts is important for our topic, since, when auditing, one does not go looking for 'waivers of responsibility', but instead one tries to find evidence, such as 'receipts created by procedure P'. In an electronic trading environment, messages have replaced the physical evidence of paper receipts, but their function remains. In other words, aligning events with economic status updates, is nothing but establishing constitutive rules, about what counts as evidence of a transaction.

2.3 Technological and Physical Infrastructure

The technological and physical infrastructure involves hardware like gates, fences, doors, locks and keys, ID passes etc. Such measures generally alter the environment in order to enforce some deserted behaviour. In general, organizational or procedural control measures like supervision make no sense, unless there is an environment containing necessary physical control measures. This holds in particular for security measures concerning the flow of goods. For instance, suppose that an radio frequency identification (RFID) reader should scan all boxes before they are cleared out of a warehouse to be exported. Installing such an RFID reader is useless when the goods are not guaranteed to pass through it. This can be achieved for instance by a system of conveyer belts. Similarly for people, a card reader for verifying ID cards makes no sense without additional gates and fences to make people pass through the gate. Physical security measures can be preventative (fence) and detective (camera systems; alarms). In both cases they help to maintain the integrity of the flow of goods, by ensuring that only authorized people may get access to the goods. The assumption is that authorized people will generally perform their duties; this assumption is warranted because all participants are certified AEO traders.

2.4 Database Management

At the data level, integrity is mostly maintained by features of the database management system (DBMS). Chen [4] distinguishes *domain integrity*, which requires that only syntactically and semantically correct data should be inserted in the database, and *referential integrity*, which implies that: "a piece of data cannot exist or be modified unless some precursor data value exists or some action is taken" [4]. A DBMS uses *integrity constraints* to verify the data and operations on the data [11]. These constraints are rules that define semantic properties to be satisfied by the data in the database in order to settle static integrity. Data is altered by updates, which together correspond to a database transaction. When regarding dynamic integrity, two things must be verified for any authorized database update. First, any newly entered input data must meet the applicable integrity constraints. Second, the transaction will guarantee that data will remain well-formed and integrity constraints are preserved [5]. Other traditional measures for maintaining data integrity are the mirroring technique, the checksum technique, the hamming codes method, and the hash function method [9]. These techniques are not deployed in a vacuum. There should be procedures to make sure these data integrity preserving techniques are not circumvented. In particular, users may only access and manipulate data through certified software applications [5]. This explains the high emphasis on change management procedures in modern information security.

2.5 Business Process and Transactions

Where do these integrity constraints come from? The answer is provided one level upwards, namely at the business process level. At the heart of every business lies the transaction. A transaction is a set of contracting relationships among individuals. It regulates an exchange of objects of value, for example money for goods. In order to perform a transaction, certain operational activities must take place. One could say that the transaction is *being constituted* by these activities. Conversely, using the terminology defined in Section 2.2, these operational activities *count as* a valid transaction. These operational activities also generate evidence which constitutes the audit trail. A transaction involves commitments by at least two parties. In all cases, some form of explicit confirmation is required to signal acceptance of a commitment. For example, an offer is responded to by a purchase order, which is subsequently accepted. After execution, again some confirmation is needed to signal acceptance of delivery [7,24]. For example, when signing for receipt of the goods (quittance), you indicate that the responsibility for the goods is transferred. Therefore confirmation evidence, like receipts or quittance, is crucial in establishing the economic status of the goods.

Business reality can be modeled as a value cycle: an interrelated system of flows of money and goods [2,20]. So-called accounts represent a state of a certain value to the company, such as inventory or accounts payable. An event is a change of state. For example, a sales event leads to an increase in accounts receivable and a decrease in inventory. Similarly, a purchase leads to an increase in inventory and an increase in accounts payable, etc. In addition, we have continuity equations for each of the states: $begin - end + in - out = 0$. A proper financial administration demands stable *reconciliation relationships* between these states or accounts. Because of the security aspect of the domain (customs inspections) we are especially interested in the completeness assertion: all goods being transported must be accounted for. Therefore, we believe that the Dutch tradition of owner-ordered accounting theory provides a good set of candidate reconciliation relations or continuity equations [22]. Unlike the U.S. manager-ordered accounting tradition, which mainly focuses on accuracy, owners are also interested in the completeness of stated revenues. In the owner-ordered tradition, there are various techniques of verifying completeness assertions by finding countervailing registrations to compare them against, see [2].

Recalculation and verification of the reconciliation relations on the basis of independent evidence is a strong detection measure to identify misstatements.

That is essentially what an auditor or tax inspector does. But crucially, such verifications can also be built into the business processes and the software that supports it. This approach is called *compliance by design* [10].

3 Illustrative Cases

In the two cases that are presented we focus on the interrelationship between external validity and internal validity. Firstly, the integrity approach will be illustrated in a case concerning pre-announcements and biometric authentication and secondly it will be illustrated in a case where electronic seals are used.

3.1 E-Link Project at Schiphol Airport

Air cargo is a fast business. Every minute counts. Given the tight schedules of arriving and departing aircraft, delays or even uncertainties lead to losses. The E-Link project is trying to reduce paperwork and waiting times of air cargo handling at Schiphol Airport, using biometrics authentication and a system of pre-announcements[1]. The scenario makes use of the identity pass for truck drivers issued by Air Cargo Netherlands (ACN). Before joining ACN a cargo handler must be certified. For more information about the project we refer to [17]. The E-Link scenario of air cargo delivery by truck to ground handlers at Schiphol Airport, runs as follows (Figure 3):

Step 1. The company sending the goods settles the commercial transactions and prepares the necessary freight documentation (air way bill, etc.).

Step 2. The company pre-announces the visit, entering all required information into the secure E-Link environment. Using this information, the security authorities can validate beforehand that the driver and truck are indeed registered to that company. Based on the pre-announcement, customs officers make a risk assessment, whether to issue an inspection or not. The results of the risk assessment are secured and unknown to the company.

Step 3. The truck arrives at the E-Link gate, the driver identifies himself with the ACN identity pass. The E-Link equipment uses biometrics (hand palm recognition) to authenticate the driver. Moreover, license plate recognition is used to authenticate the truck. Subsequently, the truck is weighed. The system verifies whether the driver, truck, company and estimated arrival time match with the pre-announcement. Afterwards, the result of the customs' risk assessment is made known: the driver is shown a display with the loading dock number to go to, or else to go to the customs inspection lane.

Step 4. At a service desk, the driver is authenticated and shows the freight documents. In the future this step can also be removed.

Step 5. At the designated loading dock the goods are unloaded. Driver, truck and freight documentation (bar code or RFID) together identify the freight.

Step 6. After unloading, a representative of the ground handler signs off for delivery of the freight. To do so, he uses his ACN pass to identify himself.

[1] `http://www.acn.nl/projecten/e-link.aspx`, last visited 8th of February 2012.

Step 7. Finally, the truck must again pass the gate, where it is weighed, and the net weight is compared to the expected weight. Before allowing the truck to leave, the system verifies that all possible outstanding issues, e.g., about differences or misunderstandings have been accounted for.

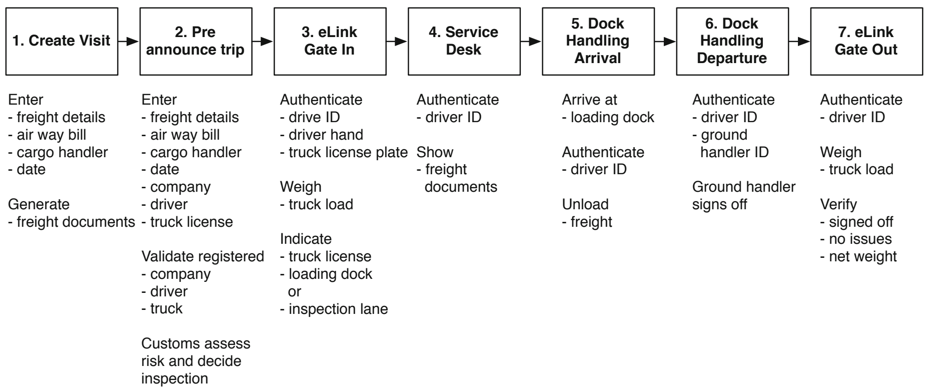


Fig. 3. E-Link process in a nutshell

The case study shows how an electronic procedure can replace the function of a traditional paper process. For example, the ground handler signing off for delivery of the goods remains as before. However, the tools are different. Instead of pink slips, now swipe cards are used in addition to biometrics authentication.

The case study also illustrates our working hypothesis, that the physical flow of goods can be reliably aligned with status updates, represented by trade documents in a shared repository, provided three conditions are met: closed environment, internal consistency and external consistency.

First, E-Link makes use of the trusted trader concept. All participants are members of ACN and must be certified by the customs authorities. Because partners guard their professional reputation at Schiphol Airport, the system of pre-announcements can be trusted. A cargo handler who would abuse the system, would surely be kicked out of the ACN alliance, meaning it is 'out of business'.

Second, The E-Link system uses several kinds of verifications and reconciliations to establish internal consistency. In particular, the system compares data on the commercial trade documents with the pre-announcement. It also compares the listed driver, truck license number and company with a database of companies with registered employees and trucks. It verifies the listed value and amount of goods with common averages for those goods. Moreover, it verifies that the ground handler has signed off for delivery. In this way, the system can establish that either a well-formed delivery transaction was performed, or else, that both parties acknowledge that there are outstanding issues to be resolved.

Third, the system contains several verifications and reconciliations to establish external consistency. In particular, the system is based on a triangulation of three data elements: the *driver*, identified by his ACN pass and authenticated by a biometrics device, the *truck*, identified and authenticated by license plate number and the *freight*, identified by the freight documentation (bar code, RFID). As additional circumstantial evidence, the truck is being weighed, both at arrival and departure. The pre-announcement is compared to the actual arrival: expected arrival time, driver, truck and freight must all correspond. If not, the goods are not allowed. At the loading dock, the system again uses identification of a person, the representative of the ground handler, as a proxy for verification of the goods. Given the trusted trader concept this assumption is warranted.

The new process leads to much improved cargo handling. First, it reduces waiting times. Now all arrivals are scheduled in advance, so there will always be enough ground handling capacity. Second, a lengthy customs verification procedure has been removed. Using the pre-announcement, customs can perform their risk assessment off-line and beforehand. In addition, the uncertainty about arrival is reduced, so customs can also schedule their inspections. In addition to these efficiency gains, also the quality of the data has improved. The number of errors and possible exceptions is reduced.

3.2 Electronic Seals

In this section we explore the possibility of electronic seals (eSeals) to help establish an integrity relationship between the goods flow and the data flow. eSeals have been studied in the Smart-CM research project [15].

Seals are ancient devices attached to the lock of a container or the lid of a box. To open the container, first the seal must be broken. Although seals can be forged, an unbroken seal counts as physical evidence that a box or container was not opened illicitly. Earlier we discussed the use of a quittance or receipt as evidence that custody over a freight is transferred. Seals can be seen as a physical counterpart to a quittance. Given the countervailing interests of subsequent parties in a supply chain, quittance will only be given, when the party receiving the goods can be reasonably sure that the box contains the goods as stated in the transport documents. When the seal is intact, this provides evidence that the box was not manipulated since it was closed. This ancient function of seals can be greatly enhanced by modern techniques.

eSeals are a particular category of smart seals that are based on active, passive or mixed RFID technology. The internal memory of the RFID chips can save a log of events. It is possible to assess the correct closure of the seal to avoid fake installations of the seal [18]. An eSeal can store in its memory if, by whom and at what time it has been opened and closed correctly. Furthermore, with active RFID technology the eSeal can communicate its state to inspectors. Therefore inspections will depend less on the inspector's skills and experience. Active RFID enables remote and on-the-fly inspection of a box or container in transit. The freight need no longer be stopped. However, as eSeals based on active RFID are battery operated, the consignor must make sure that the on-board battery can keep the eSeal alive until the end of the journey [18].

In summary, the following three kinds of security features can be exploited by the utilization of eSeals: (1) the possibility to check for proper closure of an eSeal and to save the operator identity and time of closure in its internal memory, (2) the recording of every (un)authorized operation made on the seal, along with time and date, and (3) the checking of the seal status by remotely gaining access to the seal's memory [18].

We will now review the three components for establishing integrity.

First, the trusted trader concept is used. The World Customs Organization's (WCO) Standards to Secure and Facilitate Global Trade (SAFE) Framework [23] mentions that the integrity of a box or a container brings no value to inspectors if the parties providing the integrity status are not accredited. Companies with the AEO status are considered trustworthy partners for customs authorities in Europe. Companies with the Customs-Trade Partnership Against Terrorism (C-TPAT) status are considered trustworthy partners in the USA. Both AEO and C-TPAT certificate holders are audited on compliance with security and other customs relevant standards. In other words, AEO and C-TPAT accreditation gives some assurance to customs that the entry or exit declaration data matches the true content of the container. In addition, through AEO and similarly accredited systems, a bilateral agreement exists between the involved inspection authorities and the accreditation systems are mutually acknowledged [15]. An alternative and stronger way of assuring that a container is closed by a trustworthy party is, for instance, letting it be closed under the local customs authorities' supervision [15]. Nevertheless, the identification of the party that has closed and sealed a box or a container remains a key element to establish a proper integrity relationship between the goods and the data.

Second, the tracking and tracing facilities of eSeals make it possible to establish a virtual secure pipeline, within the otherwise untrusted general supply chain. For example, in a pilot study, the beer brewer Heineken has demonstrated that it can be fully 'in control' of the export of premium beers by using electronic seals [21]. In general, to utilize eSeals, the security status information from the eSeals must be uniquely linked to cargo data. Once established, an integrity relationship between goods and data remains intact as long as the parties that close, seal and open a container are either trusted or are public inspection authorities. As it is possible to save the identity of the operator who has opened or closed an eSeal, relationships can be formed between operators closing the seals and operators opening the seals. In particular, an eSeal that is attached to a package (a box) packed inside another sealed package (a container), can only be opened after the seal on the outer container is opened. This also means that operators who are authorized to open eSeals on boxes that are aggregated in a container, should in principle also be authorized to open (and re-seal) that outer container. This implies that the order in which seals of aggregated packages are opened, should be the reverse of the order in which the seals of those packages are closed.

Third, establishing external consistency is the whole point of having seals. The crucial event is the closing or opening of a package. Therefore, there are strict procedures about opening or closing a seal. Because the procedures of parties are being audited, as part of the AEO or C-TPAT accreditation schemes, we can be reasonably certain that those procedures are being observed.

4 Discussion

In the paper we have described theoretical approaches to make sure that trade documents in a virtual data pipeline correspond to reality. In practice, adopting and implementing such measures is complex. The development is likely to meet a number of legal, commercial, political and technical challenges. Therefore, we will now briefly discuss two challenges and possible ways to overcome them.

Challenge 1. Disrupting the process. Most of the control measures we have described are not part of the primary process as they are added. However, control measures should not interfere too much with the continuation of the supply chain, otherwise they are circumvented. The flow should not be disrupted.

Consider the following example. At a loading dock in a harbor, cranes are equipped with a weighing device, which automatically weighs the container. The following control is implemented: when the weight exceeds the maximum load, the crane will stop working. The idea is that too heavy containers will affect the safety of the loading process and the ship. However, in the experience of customs officers, the control is generally switched off. A delay is simply too costly.

Suppose the controls were set up in a different way, where the crane registers the weight of each container and sums the total weigh of a freight. Now, both cargo handler and shipping company have an independent number to verify the accuracy of the trade documentation. It can also be used by customs' officers: excess weight requires some explanation. However, importantly, such differences can be settled later and will not disrupt the loading process.

Challenge 2. Information quality and role in the chain. Customs have the duty to stop undesired goods like drugs, weapons, or counterfeit goods at the border (stop function). The customs office would like to use the virtual data pipeline for targeting its inspections. Nowadays, customs use a risk-based inspection approach. Based on stated properties of the goods, the shipper, the country of origin and destination, etc., certain freights are selected for inspection. However, this requires that the data are accurate and complete.

However, currently the data in customs declarations is of low quality. The data are incomplete, late, or no longer accurate after re-routing. Customs declarations are filled in by the carrier. However, the carrier is usually not in a position to provide more accurate and detailed information; such information is not needed for fulfilling his role in the supply chain. What is more, telling the carrier about the contents of boxes would increase the likelihood of theft. Participants will be very reluctant to share such information.

Therefore, supply chain visibility for regulatory purposes can only be a success, when the legal responsibility to fill out the details of the goods is somehow transferred to the supply chain as a whole, involving the consigner and the consignee. This may be very difficult to achieve from an administrative point of view. Moreover, to make the concept of a secure pipeline acceptable for commercial parties, it should be possible to demonstrate that parties will only get access to shared information in the pipeline, which is needed for their task in the chain (principle of least privilege).

5 Conclusions and Future Research

In this paper we have looked at supply chain visibility: reusing commercially available data to trace goods along the supply chain and make them visible for regulatory purposes. A particular approach is the virtual data pipeline concept: an infrastructure for exchanging trade documents among trusted traders [14]. For successful adoption, trust in the reliability of the data in the virtual pipeline is crucial. In this paper, we have looked at theories about the way in which the trade documents which are being being exchanged in the virtual data pipeline can be undeniably linked to the physical flow of goods.

We argued that in order to do so, we need three kinds of control measures:
(i) All participating companies are known and certified (trusted traders). This warrants the assumption that participants' employees can generally be trusted to do their work, for instance, to apply seals or sign off according to procedures.
(ii) Measures to establish internal validity. Database management systems can maintain integrity constraints for the data being stored. The integrity constraints are derived from the meaning of the documents in the trade process. All steps in the process are interpreted as status updates concerning the exchange of value, of goods, or of information. These steps in the process should together form a well-formed transaction, from an economic point of view. The system can verify specific reconciliation relations, derived on the basis of accounting theory.
(iii) Measures to establish external validity. All control measures are rendered powerless, unless there are basic physical security measures in place, like fences, doors, locks and keys. These measures protect the integrity of a storage space, such as a warehouse, truck or container. In addition, there are modern technological control measures, like authentication devices (biometrics) or electronic seals (RFID), which have taken over the function of their ancient and proven predecessors, but have advanced them. In particular, these new technologies make it possible to trace the goods or the people handling them from a distance.

The control measures are linked in a three layered architecture, consisting of the business process or value layer, the information or data layer and the physical infrastructure layer. The relationship between layers is a representational one: events or facts at the physical layer generate evidence which *counts as* or *constitutes* evidence of institutional events or facts at the business layer.

The approach is illustrated by two cases. The E-Link project about air cargo export at Schiphol Airport illustrates that it makes sense to share information early on, and thereby reduce waiting time and uncertainty. The electronic seals developed in the Smart-CM project show that much security related supply chain information can be stored locally in the seals themselves which makes the chain transparent. In future research we will investigate this line of reasoning further and show how to model and formalize a specific set of continuity equations for each of the layers, and align them, for a specific non-trivial case.

Acknowledgments. We are grateful to our colleague Yao-Hua Tan for valuable input. We would like to thank Aike Müller for information about the E-Link project at Schiphol airport.

References

1. Aldewereld, H.: Autonomy vs. Conformity - An Institutional Perspective on Norms and Protocols. Ph.D. thesis. Universiteit Utrecht (2007)
2. Blokdijk, J., Drieënhuizen, F., Wallage, P.: Reflections on Auditing Theory, A contribution from the Netherlands. Limperg Institute Amsterdam (1995)
3. Boritz, J.: Is practitioners' views on core concepts of information integrity. International Journal of Accounting Information Systems 6(4), 260–279 (2005)
4. Chen, S., Drake, J., Tsai, W.: Database requirements for a software engineering environment: criteria and empirical evaluation. Information and Software Technology 35(3), 149–161 (1993)
5. Clark, D., Wilson, D.: A comparison of commercial and military computer security policies. In: IEEE Symposium on Security and Privacy, pp. 184–194 (1987)
6. European Commission: AEO guidelines. Tech. Rep. TAXUD/2006/1450, Directorate-General Taxation and Customs Union (2007)
7. Dietz, J.L.: The deep structure of business processes. Communications of the ACM 49(5), 59–64 (2006)
8. FIPS: Minimum security requirements for federal information and information systems. Tech. rep., U.S. National Institute of Standards in Technology, NIST (2006)
9. Ghaeb, J., Chebil, J.: An oblique-matrix technique for data integrity assurance. Computers & Security 28(1-2), 94–99 (2009)
10. Governatori, G., Sadiq, S.: The journey to business process compliance, pp. 426–445. IGI Global (2009)
11. Grefen, P., Apers, P.: Integrity control in relational database systems - an overview. Data and Knowledge Engineering 10(2), 187–223 (1993)
12. Grossi, D.: Designing Invisible Handcuffs: Formal Investigations in Institutions and Organizations for Multi-agent Systems. Ph.D. thesis. Universiteit Utrecht (2007)
13. Hesketh, D.: Weaknesses in the supply chain: who packed the box? World Customs Journal 4(2), 3–20 (2010)
14. Hofman, W.: Supply chain risk analysis with linked open data. In: Vermaas, P.E., Dignum, V. (eds.) FOMI 2011, pp. 77–87. IOS Press, Amsterdam (2011)
15. INTEGRITY, SMART-CM: Global container supply chain compendium (2008)
16. Klein, R., Rai, A.: Interfirm strategic information flows in logistics supply chain relationships. MIS Quarterly 33(4), 735–762 (2009)
17. Müller, A.: Re-engineering Security: A risk based approach for information security focused business process re-design. Master's thesis. Erasmus Universiteit (2012)
18. Rizzo, F., Barboni, M., Faggion, L., Azzalin, G., Sironi, M.: Improved security for commercial container transports using an innovative active RFID system. Journal of Network and Computer Applications 34(3), 846–852 (2011)
19. Searle, J.R.: The Construction of Social Reality. The Free Press (1995)
20. Starreveld, R.W., de Mare, B., Joels, E.: Bestuurlijke Informatieverzorging, vol. 1. Samsom, Alphen aan den Rijn (1994) (in Dutch)

21. Tan, Y., Bjørn-Andersen, N., Klein, S., Rukanova, B. (eds.): Accelerating Global Supply Chains with IT-Innovation. Springer, Berlin (2011)
22. Vasarhelyi, M.A., Alles, M., Kogan, A.: Principles of analytic monitoring for continuous assurance. J. of Emerging Technologies in Accounting 1(1), 1–21 (2004)
23. WCO: SAFE framework of standards to secure and facilitate global trade (2011)
24. Weigand, H., de Moor, A.: Workflow analysis with communication norms. Data and Knowledge Engineering 47(3), 349–369 (2003)
25. Yu, L.: Linked open data. In: A Developer's Guide to the Semantic Web, pp. 409–466. Springer, Heidelberg (2011)

An Empirical Assessment of the Effect of Context-Based Semantic Annotation on Process Model Discovery

Mturi Elias and Paul Johannesson

Department of Computer and Systems Science (DSV),
Stockholm University (SU) Stockholm, Sweden
{mturi,pajo}@dsv.su.se

Abstract. CPSAM is a context-based process semantic annotation model for annotating business processes in a process model repository. The purpose of the annotation model is to facilitate searching process models, navigating a process model repository and enhance users' understanding of process models. The annotation model has partly been evaluated through an empirical study to test the annotation consistency and correctness. The purpose of this paper is to examine the effect of the process annotation (based on CPSAM) through a controlled experiment, where a prototype of the repository is used, to annotate and store process models based on the CPSAM. The evaluation is supposed to test whether process annotation based on the CPSAM can facilitate searching, navigation and understanding of process models stored in a repository. The results show that annotating business processes using the annotation model positively affects searching process models, navigating the repository and understanding process models.

1 Introduction

Business Process Management (BPM) has become one of the most important instruments that help modern organizations meet their business goals and achieve competitive advantage. Business process modeling plays a vital role in BPM. Motivations for modeling business processes include documenting current business processes, redesigning and improving processes, aligning business and IT, etc. Modeling of business processes is a complex, costly and time consuming task [1, 2], however, the efforts made to model business processes are seldom reused beyond their original purpose. An attractive alternative to modeling business processes from scratch is deriving them by redesigning existing models. Such an approach requires the use of business process model repositories that provide a location for storing and managing process knowledge for future reuse.

There exists a number of efforts to build process model repositories, e.g. the MIT Process Handbook [3], SCOR [4], SAP's Business Map [5], and IBM's Patterns for E-Business [6]. However, the use of such repositories is still limited and fragmented [7]. In order to investigate the reasons for this limited use, a number of requirements on business process repositories were elicited in our previous study [8]. These

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 366–382, 2012.

requirements were then used to review and compare a number of existing repositories [9]. A main finding was that any repository requires effective instruments for searching and navigating its contents. In order to address this challenge, a context-based process semantic annotation model (CPSAM) [10] was developed for semantically annotating business processes in the repository. The purpose of the annotation model is to facilitate searching process models, navigating the repository and enhance understandability of process models. The annotation model (CPSAM) consists of the following annotation elements, *process type*, *process area, resource*, *actor*, *organizational level*, *process phase*, *process relationship*, *business context*, and *goal,* as shown in Figure 1.

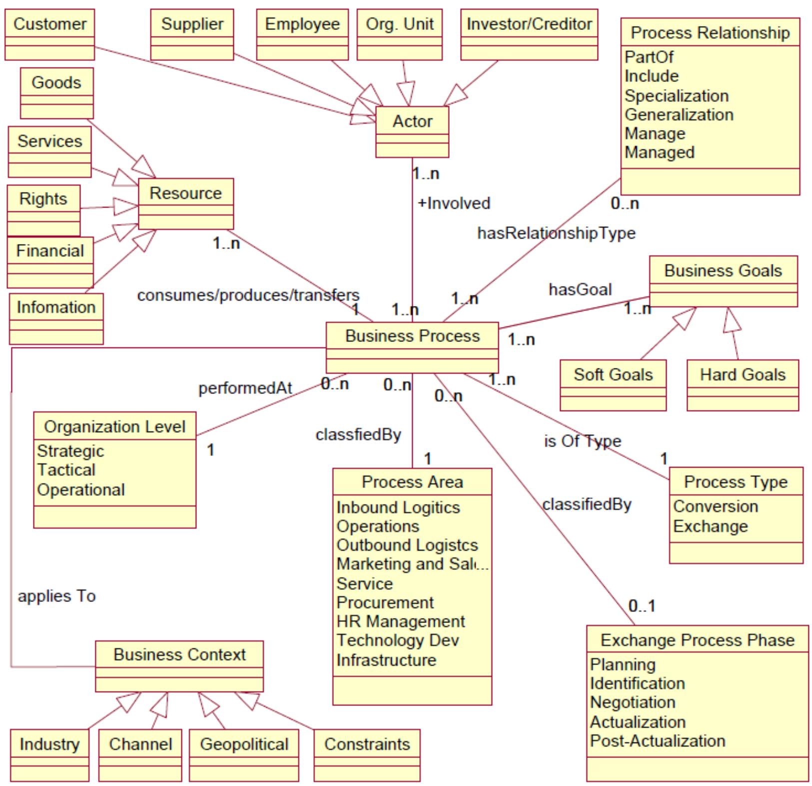


Fig. 1. A Context-based Process Semantic Annotation (CPSAM) [10]

- The *business process* is the main concept that is described. It captures some basic information such as version number, process name, etc.
- *Process Area* element classifies business processes by their function in the value chain or core competence.
- The *organizational level* element describes the level in the organization at which a business process is performed.

- The *process type* element classifies business processes according to whether they aim to exchange or transform resources.
- *Exchange Process Phase* element classifies processes based on activities needed for exchanging resources that extend over several phases from initial planning to follow-up activities after the actual exchange has been completed.
- *Resource* describes the resource that is being consumed/produced by a conversion process or exchanged by actors in an exchange process.
- *Actor* describes an entity such as a person or an organizational unit involved in the realization of a business process.
- The *process relationship* element describes how business processes are related.
- A *business context* defines the circumstances in which a process may be used.
- The *business goal* element describes the business goals that a process model is intended to achieve.

The annotation model extends existing process classification schemes, [11, 12] by incorporating elements from well established frameworks in accounting [13, 14], organizational theory [15, 16], and enterprise modeling, [17]. A detailed description of the annotation model can be found in [10].

The CPSAM was evaluated in our previous study [10], however, the scope of the study was limited to evaluating the consistency and correctness of annotating business processes by using CPSAM i.e. the model has not been evaluated to measure its objective (searching, navigation and understandability). Therefore, the aim of this study is to evaluate the effect of the context-based process annotation through a controlled experiment to measure its performance and users' perception.

Against this background, the remainder of the paper is structured as follows. In Section 2, the exemplar application of the annotation model using a repository prototype is presented. Section 3 introduces the evaluation framework and the experimental settings. In Section 4 we present the results and discussion of the study. Finally, the paper is concluded in Section 5.

2 Exemplar Application of the CPSAM

In this section we demonstrate the use of the implemented Context-based Process Semantic Annotation Model in the repository. For the demonstration, an order-to-cash business process is used as a running example. An order-to-cash is the business process where goods are: ordered, delivered and received, as well as invoiced and paid for. All order-to-cash processes include activities related to invoicing, delivery and payment, however, they have several differences. For example an order-to- cash process for the delivery of goods (e.g., personal computers) is different from the one for the delivery of services (e.g., consultancy services). The use of CPSAM for classifying and describing these processes captures their similarities and differences, thus enabling a repository user who is searching for one of the two processes, to find a relevant process model.

An example scenario starts with a business analyst who uses the tool to annotate an order-to-cash process model for the delivery of goods, and stores it in the repository. The scenario is followed by another user who performs a search in the repository to find an order-to-cash process model for delivery of service.

Task 1: Annotating a Process Model

Suppose the analyst has already designed an order-to-cash process model shown in Figure 2 (a). Before storing the process model, the analyst is required to annotate the process model using the annotation service provided in the repository. To ensure correct annotation, the help service in the repository provides definitions of each annotation element. Thus the analyst annotate a process model as follows:

- The analyst is the process owner, so he provides the *process name* and *process description* of a process model. The *version number* is assigned automatically.
- The main activities of this process regards getting buyers to purchase products i.e. selling, therefore the “Process Area” is “Marketing and Sales”.

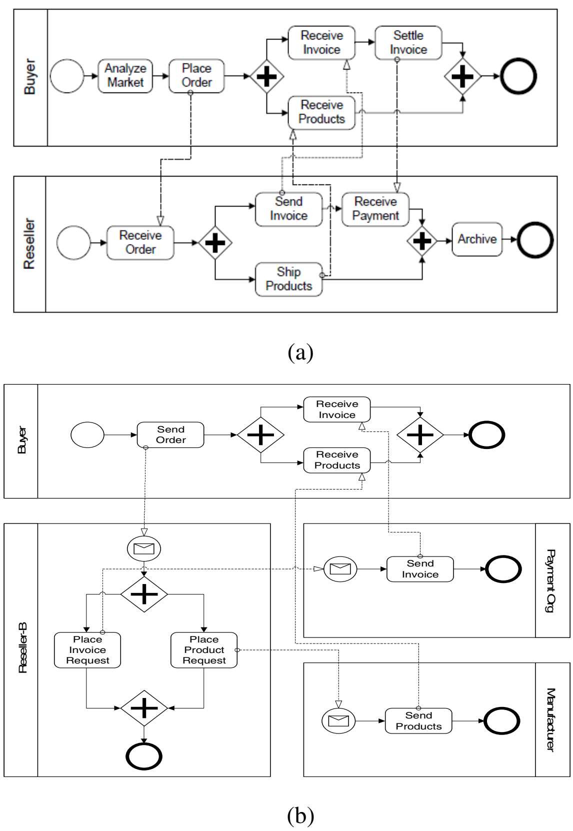
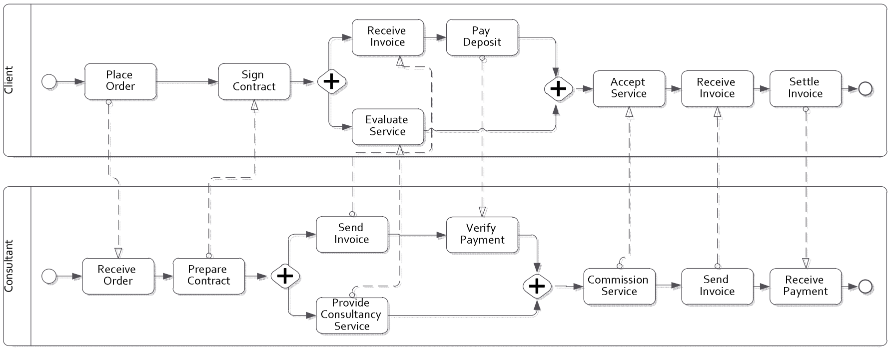


Fig. 2. Order to cash process models



(c)

Fig. 2. (*Continued*)

- It includes day-to-day activities, therefore "Organization Level" is "operational"
- The process involves exchange of resources, hence "Process Type" is "Exchange"
- The process includes activities for preparing and performing the exchange, therefore "Exchange Process Phase" is "Actualization".
- The actor involved in the exchange identified as "Principal Actor" is "Supplier", whose role is a reseller and "Other Actor" is "Customer", whose role is a buyer.
- The reseller receives payment (i.e. cash, cheque) and ship products (goods) to the buyer, therefore the resource being exchanged identified as "Resource Received" is "Financial" and "Resource Provided" is "Goods".

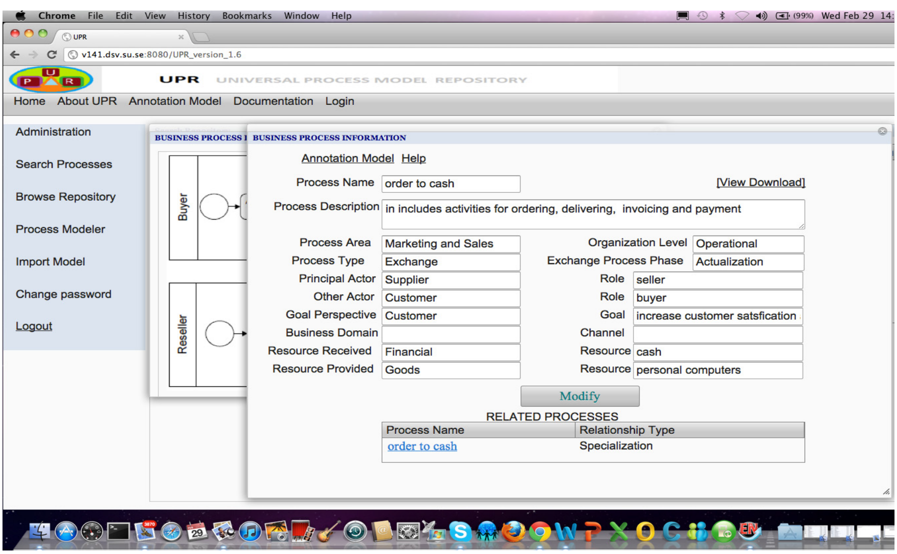


Fig. 3. Annotation of an order to cash process model (for delivery of goods)

- The goal of this process is to "Increase customer satisfaction and retention" and "Goal Perspective" is "Customer"
- This is a generic order to cash process, not restricted to any domain; therefore it is not annotated with business context information.

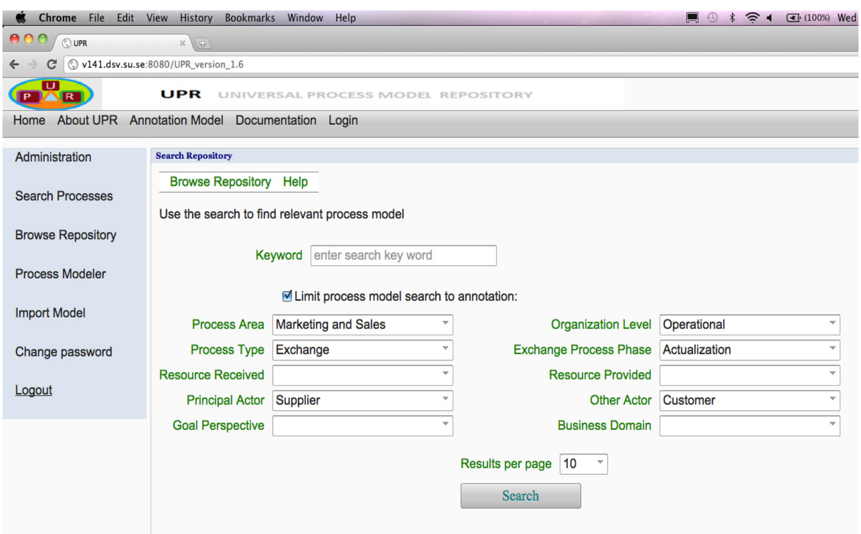
The analyst will produce the annotation as shown in Figure 3 above.

Task 2: Searching a Process Model

In this task we describe how a repository user uses the CPSAM based annotation to search for an order-to-cash process model (for delivery of service) in the repository.

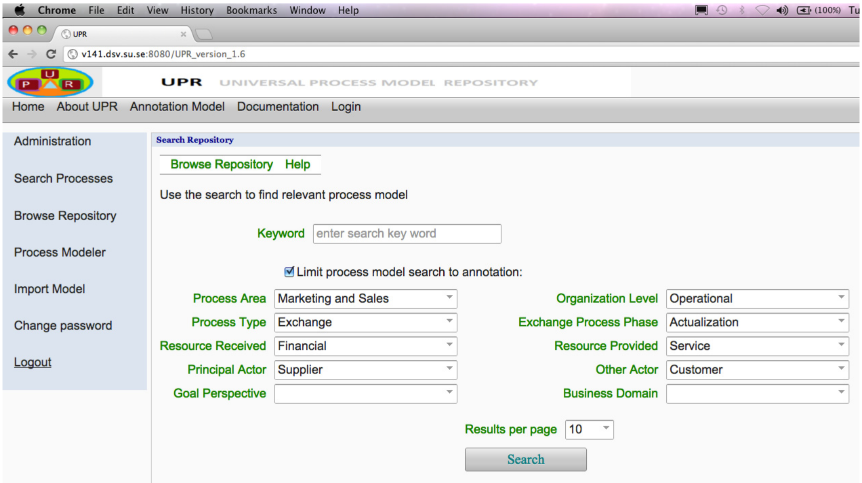
Using annotation-based search, the user looking for an order-to-cash process model for consultancy service delivery, can limit the search to annotation elements. Starting with *process area, organization level, process type, and process phase, principal actor and other actor* to form query 1 shown in Figure 4 (a), the query retrieves the three process models in Figure 2, from the studied repository. Therefore, a user can take a strategy of stepwise narrowing the query by using more annotation elements. Since the user is looking for an order to cash process model for service delivery, this means the *resource provided* in exchange is 'Service' and the *resource received* is 'Financial'. In this way, we decrease the search space to form query 2 in Figure 4 (b), and the process model in Figure 2 (c) is retrieved as relevant.

A query may result in any number of process models that are considered as relevant. Therefore, the user may identify the relevant process model that meets his or her business need from the retrieved models. Otherwise, the user may decide to narrow or expand the query for further search.



(a) Query 1

Fig. 4. Semantic-based Process Model Search



(b) Query 2

Fig. 4. (*Continued*)

3 Evaluation Framework and Experimental Settings

Our evaluation framework follows the Method Evaluation Model (MEM)[18], a method for evaluating IS design methods. The MEM is based on two areas of theory: the Technology Acceptance Model (TAM) [19], from the IS success literature and Methodological Pragmatism [20] from the philosophy of science. The MEM is chosen because it incorporates both aspects of evaluation i.e. performance, and user perception. Figure 5 shows the MEM indicating the primary constructs and causal relationships between them. The construct of the MEM in the context of CPSAM can be defined as follows.

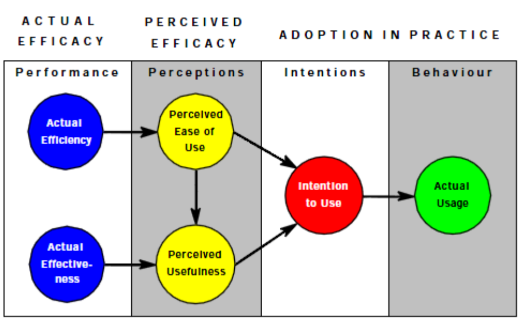


Fig. 5. The Method Evaluation Model (MEM) [18]

Actual efficacy is the degree to which the use of CPSAM based annotation achieves its objectives, which in this case are searching process models, navigating process model repository and understanding process models. *Actual efficiency* can be measured by investigating the extent to which the use of the CPSAM based annotation reduces the time or effort required to search and navigate the repository to find a relevant process model, as well as the effort or time required to comprehend a process model. *Actual effectiveness* can be measured by investigating the extent to which the use of CPSAM based annotation improves the quality (i.e. accuracy) of the search, navigation and comprehension of process models.

Perceived ease of use is the degree to which a person believes that using the CPSAM based annotation in searching, navigating and comprehending process models is free of effort.

Perceived usefulness is the degree to which a person believes that the use of the CPSAM based annotation will be effective (i.e. accurate) in searching, navigating and comprehending process models in the repository.

Intention to use is the extent to which a person intends to use the CPSAM based annotation for searching, navigating and comprehending a process model.

3.1 Experimental Settings

The purpose of this study is to test whether the annotations based on CPSAM meets its objectives—to facilitate searching of process models, to support navigation of the repository and enhance understandability of process models. In this research we have carried out a two-stage evaluation test: In the first stage, we used a controlled experiment to test the performance (actual efficacy). For that purpose, we formulated the following hypotheses for measuring the performance of the annotation model on searching, navigation and understandability of process models.

H1: The annotation positively affects searching of process models in the repository
H2: The annotation positively affects navigation of the process model repository
H3: The annotation positively affects process model understandability

In the second stage, we have used a survey (using a questionnaire) to test the user's perception on the effect of annotation. In the following subsections, a detailed description of the experimental design is presented.

Experimental Materials and Tasks. The main instrumentation for the experiment was a repository prototype demonstrated in section 2. For the experiment, the repository was populated with more than 100 business process models adopted from existing repositories, i.e. MIT, IBM and SAP. The process models were redesigned using BPMN, a standard process modeling notation, and stored in the repository. In addition to that the following materials were prepared for the experiment

- A document describing the CPSAM annotation model
- A document describing the prototype of the repository

- A post-task survey questionnaire to measure the user perception of the effect of the annotation. The survey consisted of eight closed questions assessed over a scale of 1 to 5 (Strongly Disagree, Disagree, Not Sure, Agree, and Strongly Agree)
- A document describing a set of questions for each experimental task. The document included five questions for searching, five questions for navigation, and four questions for understanding i.e. one question for each process design perspective (functional, behavioral, organizational and informational).

The questions related to the understanding task were accompanied with six process models (P1 to P6), where half of the accompanied processes were annotated and the others were not. During the experiment the participants were divided into two groups. The first group of the participants was given P1, P3 and P5 as annotated process models, whereas the second group was given P2, P4 and P6 as annotated process models.

The experiment consisted of four main tasks, which are as follows:

- *Searching Task.* In this task participants were asked to find process models that are relevant to a given question. For each question, participants were required to perform both the keyword-based and the annotation-based search. From questions, participants were supposed to identify some keywords and annotations that they would use for searching.
- *Navigation Task.* In this task participants were asked to navigate the repository and locate process models that are relevant to a given question. For each question, participants were required to perform both the alphabetical-based and the annotation-based navigation. From questions, participants were supposed to identify some alphabets of the keyword and annotations that would guide them in navigating the repository.
- *Understandability Task.* In this task participants were required to study process models and answer related questions. As discussed above, half of the process models were annotated and other half were un-annotated.
- *Post Task Survey.* Upon completing the experiment participants were asked to perform a post task survey.

Participants Selection and Experimental Treatment. The participants involved in the experiment were a mix of Masters students in Engineering and Management of Information Systems (EMIS) and PhD students in Information Systems at KTH. By the time the experiment was done, all students had completed a course on Enterprise Systems and Modeling, in which they learnt basic concepts about business process modeling. The benefit of using student participants is that they form a homogeneous group with respect to their academic background and industrial experience. Furthermore, the experimental tasks did not require high levels of industrial experience, which justifies our selection of the participants.

At the beginning of the experiment, the participants were given a short list of written instructions describing the experiment. Experiment mentors demonstrated how the prototype could be used to search and navigate the repository. Furthermore, a case of understanding process models annotation was demonstrated.

For the experiment, 20 randomly selected participants were given the materials (described in section 3.2). Responses from 15 participants were received and all the collected data is considered for analyses. Due to the length of the experiment each participant was asked to perform two tasks and a post task survey. For the analyses (in section 4) of each task (searching, navigation and understandability) results from 10 participants are included.

3.2 The Studied Variables

In order to test the influence of the annotations (based on the CPSAM) on searching, navigation and process model understandability, we distinguish two types of variables: performance based (objective) and perception based (subjective) measures.

3.2.1 Performance Based Variables

Variable 1*. Search Correctness (SC)*: The degree of accuracy with which a user finds a relevant process model by searching the repository. It is measured in terms of F-measure—the harmonic mean of *precision* and *recall*. *Precision* is the fraction of retrieved process models that are relevant, whereas *recall* is the fraction of relevant process models retrieved. F-measure is a standard measure for evaluating information retrieval results [21].

The steps taken for measuring *SC* are to let different participants find process model(s) from the repository that are relevant to given question and then we compute *SC* as follows:

1. Let *RR(q, i)* be the total number of relevant processes retrieved on question *q* by subject *i*, *IR(q, i)* be the number of irrelevant processes retrieved on question *q* by subject *i*, and *RN(q, i)* be the number of relevant processes in the repository that have not been retrieved. The precision *PR(q, i)* and recall *RC(q, i)* on question *q* by subject *i* is, *PR (q, i) = RR (q, i) / [RR(q, i) + IR (q, i)]* and *RC (q, i) = RR (q, i) / [RR(q, i) + RN (q, i)]*
2. The Search Correctness on question *q* by subject *i*, measured by F-measure, *F(q, i) = 2* PR(q, i) * RC(q, i) / [PR(q, i) + RC(q, i)]*
3. The average *SC* by subject *i, F(i)* = (•*F(q, i))/n* for *q* = 1 to *q* = n, where n is the number of process retrieval questions.

By comparing the results of *SC* for the keyword based and annotation based search, we can determine whether the annotation positively affect the searching or not.

Variable 2. *Navigation Efficiency (NE)*: Is the proportion of the steps (efforts) that are useful to find the relevant process models in the repository. It is measured by the minimum path length (*MPL*) divided by the total user path length (*TUPL*) [22, 23] used to locate the process model. The path length is the number of steps (button clicks) performed in order to find relevant process models by navigating the repository. The total user path length is the total number of steps a user used to locate relevant process model by navigating the repository. The minimum path length is the least number of steps needed to locate relevant process model.

The steps taken for measuring *NE* are to let participants locate a process model relevant to a given question by navigating the repository and then we compute *NE* as follows:

1. Let *MPL(q)* be the least number of steps needed to locate a process model for question *q* by navigating the repository and *TUPL(q, i)* be the total number of steps subject *i* used to locate relevant processes for question *q*.
2. The Navigation Efficiency for locating a process for question *q* by subject *i* is, *NE(q, i) = MPL(q) / TUPL (q, i).*
3. The average *NE* by subject *i*, *NE(i) = (•NE(q, i))/n* for *q* = 1 *to q* = n, where n is the number of process retrieval questions.

By comparing the results of *NE* for alphabetic-based and annotation based navigation, we can determine whether the annotation positively affect the navigation or not.

Variable 3*. Understandability (UL)*: It is the degree of correctness to which a user understands a process model. It is measured as the fraction of correct answers given by the subject to the different questions about the process [24].

The steps taken for measuring *UL* are to let subjects study a process model and respond to questions related to the process model. We then compute *UL* as follows:

1. Let *CA(p, i)* be the number of correct answers on process *p* by subject *i* and *EA(p)* be the number of expected correct answers on process *p*. The understandability on process *p* by subject *i* is given by *UL (p, i) = CA(p, i)/ EA(p).*
2. The average understandability by subject *i*, is *UL(i) = (•UL(p, i))/n* for *p*=1 to *p*=n, where n is the number of process models.

By comparing the results of *UL* for annotated and unannotated process models, we can determine whether the annotation positively affect the understandability or not.

3.2.2 Perception Based Variables

Variable 4. *Perceived Usefulness (PU)***:** Is the degree to which a person believes that the annotations (based on CPSAM) improve searching, navigation and understanding of process models.

Variable 5. *Perceived Ease of Use (PEOU)*: Is the degree to which a person believes that the use of the CPSAM based annotation is free of effort.

Variable 6. *Intention to Use (IU)*: Is the extent to which a person intends to use the CPSAM based annotation for searching, navigating and comprehending a process.

In order to investigate users' perception of the model we asked the participants to assess several statements on a scale of 1 to 5 (Strongly Disagree, Disagree, Not Sure, Agree and Strongly Agree). The statements (as shown in **Table 1**) are PU1, PU2, and PU3 for *Perceived Usefulness*, PEOU1and PEOU2 for *Perceived Ease of Use*, and IU1, IU2 and IU3 for *Intention to Use*.

Table 1. Item for measuring perception based variables

Items	Statements
PU1	I think the annotations have improved the process of locating and searching of process models in the repository
PU2	I found navigating the process model repository based on CPSAM elements to have improved my work
PU3	I found the annotations to be helpful for understanding process models
PEOU1	It was easy for me to locate/search the process models
PEOU2	It was easy for me to navigate the repository
IU1	If I have to search a process in the repository in the future I will use annotation based search
IU2	If I have to navigate the repository in the future I will use annotation based approach
IU3	If I am involved in building the repository for process models I would recommend the CPSAM model

4 Results and Discussion

In this section, the data collected from the study are analyzed and discussed in order to evaluate the CPSAM. The Wilcoxon matched pairs test is chosen for data analysis, as demonstrated by Zobel [25], Wilcoxon's signed rank test is a reliable way to evaluate statistical differences between two retrieval systems. The following are the results and discussion for each studied variable.

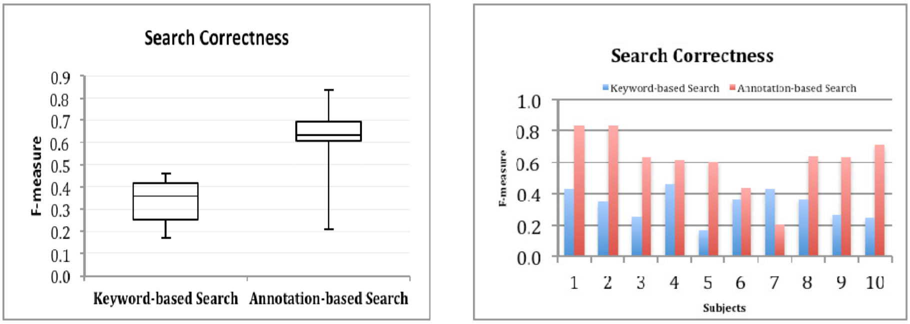
Effects of Annotation on Searching of Process Models. Figure 6 depicts boxplots (a) and graph (b) of F-measure for keyword based search and annotation based search.

Table 2. Search Correctness

Subjects	F-KBS	F-ABS	Differences	Diff. Rank	Signed Rank
1	0.430000000	0.833333333	-0.403333333	7	-7
2	0.351052632	0.833333333	-0.482280702	10	-10
3	0.251052632	0.633333333	-0.382280702	6	-6
4	0.460769231	0.617142857	-0.156373626	2	-2
5	0.167719298	0.604617605	-0.436898306	8	-8
6	0.367719298	0.440000000	-0.072280702	1	-1
7	0.434385965	0.206349206	0.228036759	3	3
8	0.367719298	0.638961039	-0.271241741	4	-4
9	0.266666667	0.633333333	-0.366666667	5	-5
10	0.246000000	0.713333333	-0.467333333	9	-9

The plot and the results in Table 2 shows that Search Correctness for annotation-based search (ABS) is better than keyword-based search (KBS) given that the median value for F-measure is higher for ABS than for KBS. This indicates that annotation positively affects the searching of process models in the repository.

Using the data in Table 2 and the Wilcoxon signed-ranks test and 0.01 significance level, we test the claim that there is no difference between keyword-based search (KBS) and annotation-based search (ABS). The sum of the absolute values of the negative ranks, T-= 52 and the sum of the positive ranks, T+=3. Because n=10, we have $n \leq 30$, so we use a test statistic of T=3. Therefore, Critical Value(0.01(2),10)=3.0 and Critical Value(0.01(1),10)=5.0 Since T+ is less than or equal to the critical value we reject the null hypothesis. Thus hypothesis H1 is accepted.



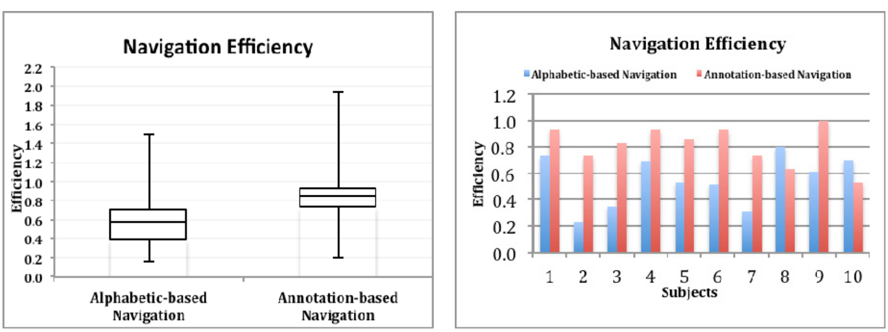
(a) (b)

Fig. 6. Search Correctness

Effects of Annotation on Navigating the Process Models Repository. Figure 7 depicts the boxplots (a) and graph (b) of navigation efficiency for alphabetic-based navigation (ALN) and annotation-based navigation (ANN).

Table 3. Navigation Efficiency

Subjects	NE-ALB	NE-ANB	Differences	Diff. Rank	Signed Rank
1	0.733333333	0.933333333	-0.200000000	3	-3
2	0.230000000	0.733333333	-0.503333333	10	-10
3	0.351904762	0.833333333	-0.481428571	9	-9
4	0.695238095	0.933333333	-0.238095238	4	-4
5	0.533333333	0.866666667	-0.333333333	5	-5
6	0.516666667	0.933333333	-0.416666667	7	-7
7	0.306666667	0.733333333	-0.426666667	8	-8
8	0.800000000	0.633333333	0.166666667	1.5	1.5
9	0.616666667	1.000000000	-0.383333333	6	-6
10	0.700000000	0.533333333	0.166666667	1.5	1.5



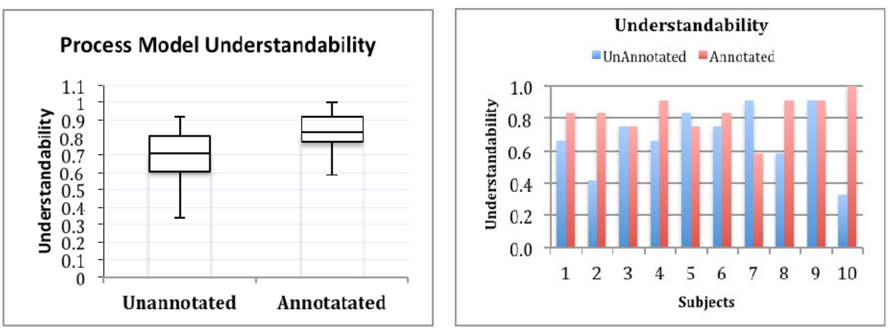
(a) (b)

Fig. 7. Navigation Efficiency

The plot and the results in **Table 3** shows that Navigation Efficiency (*NE*) for annotation-based navigation (ANN) is better than alphabetic-based navigation (ALN) given that the median value of *NE* is higher for ANN than for ALN. This indicates that annotation positively affects the navigation performance in the repository.

Using the data in **Table 3** and the Wilcoxon signed-ranks test and 0.05 significance level, we test the claim that there is no difference between alphabetic-based navigation (ALN) and annotation-based navigation (ANN). The sum (T-) of the absolute values of the negative ranks is 52 and the sum (T+) of the positive ranks is 3. Because n=10, we have n≤30, so we use a test statistic of T=3. Therefore, Critical Value(0.05(2),10)=8.0 and Critical Value(0.05(1),10)=10. Since T+ is less than or equal to the critical value we reject the null hypothesis and accept hypothesis H2.

Effects of Annotation on Process Model Understandability. Figure 8 depicts the boxplots (a) and graph (b) of understandability (UL) of un-annotated and annotated process models. The boxplots and the graph shows that understandability of annotated process models is higher than understandability of un-annotated process models, given that median of understandability for annotated models is higher than the median of understandability of un-annotated model. This indicates that annotation positively affects process model understandability.



(a) (b)

Fig. 8. Process Model Understandability

Using the collected data and the Wilcoxon signed-ranks test and 0.05 significance level, we test the claim that there is no difference between understandability of un-annotated models and annotated model. The sum of the absolute values of the negative ranks, T-= 29 and positive ranks, T+=7. Because n=8 (we omit two values with difference = 0), we have n≤30, so we use a test statistic of T=7. Therefore, Critical Value(0.05(2),8)=3.0 and Critical Value(0.05(1),8)=5.0. Since neither T+ nor T- is less than or equal to the critical value we fail to reject the null hypothesis.

User Perception on the Annotation. Figure 9 shows the summary of statistics for user perception of the annotation model.

Perceived Usefulness (PU). The graph shows that more than 80% of participants at least agree (i.e. agree and strongly agree) with PU2 and PU3, whereas, more than

60% at least agree on PU1. Therefore, it can be argued that most users perceived the annotation to be useful.

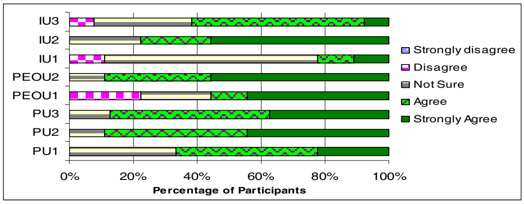


Fig. 9. User Perception of the Annotation Model

Perceived Ease of Use (PEOU). The graph shows that more than 50% at least agree on PEOU1, whereas more than 80% of participants at least agree with PEOU2. Therefore, it can be argued that most users perceived the annotation to be easy to use.

Intention to Use (IU). The graph shows that less than 30% of participants at least agree with IU1 (i.e. more than 60% of participants are not sure about IU1), whereas more than 60% at least agree with IU2 and IU3. This implies that most users are not sure about their intention to use annotation-based search. We hypothesize that the reasons to this are, a) that annotation based search requires some time and effort to think about different annotation elements before searching, b) people are used to keyword based search is their searching routine.

5 Conclusion

In this paper, we have evaluated the effect of context-based process semantic annotation through a controlled experiment to test whether annotation can facilitate searching, navigating and understanding process models stored in a repository. For the evaluation we have used the Method Evaluation Model (MEM), a widely accepted model for measuring the performance and user perception of artifacts. In order to perform the experiment we have implemented a repository prototype that implements the annotation model and populated it with more than 100 process models.

The results provide evidence that the annotation model positively affects searching and navigating a process model repository. Furthermore, the results indicate that the annotation model positively affects the understandability of process models. However, the effect of the model on understandability is not significant. One of the reasons could be that most process models used for the experiment were not complex. Therefore, we hypothesize that CPSAM based annotation could improve the understandability for very large and complex process models.

The results from the post task survey suggest that most users perceived the annotation as easy to use and useful for searching, navigation and understanding of process models. Also, the results showed that users have positive intention to use the annotation model for navigation and understanding. However, most users are not sure about their intention to use the annotation model for searching. Possible reasons may include: a) annotation based search requires some time and effort to think about different annotation elements before searching, b) people are used to keyword based search in their searching routine.

One of the limitations of the study is that a small number of participants were used for the experiment. Future research aims at a large scale evaluation of the annotation model and to improve the annotation model based on the evaluation results.

References

1. Rodrigues Nt., J.A., de Souza, J.M., Zimbrão, G., Xexéo, G., Neves, E., Pinheiro, W.A.: A P2P Approach for Business Process Modelling and Reuse. In: Eder, J., Dustdar, S. (eds.) BPM Workshops 2006. LNCS, vol. 4103, pp. 297–307. Springer, Heidelberg (2006)
2. Markovic, I., Pereira, A.C.: Towards a Formal Framework for Reuse in Business Process Modeling. In: ter Hofstede, A.H.M., Benatallah, B., Paik, H.-Y. (eds.) BPM Workshops 2007. LNCS, vol. 4928, pp. 484–495. Springer, Heidelberg (2008)
3. `http://process.mit.edu/Default.asp` (last accessed March 10, 2012)
4. `http://repository.phios.com/SCOR/` (last accessed March 10, 2012)
5. `http://help.sap.com/saphelp_sm40/helpdata/EN/5e/c8145e3a9d9340913099159d80fc87/frameset.html` (last accessed March 10, 2012)
6. `http://publib.boulder.ibm.com/infocenter/wchelp/v5r6m1/indexjsp?topic=/com.ibm.commerce.business_process.doc/concepts/processPrice_order.html` (last accessed March 10, 2012)
7. La Rosa, M., Reijers, H.A., van der Aalst, W.M.P., Dijkman, R.M., Mendling, J., Dumas, M., García-Bañuelos, L.: APROMORE: An advanced process model repository. Expert Systems with Applications 38, 7029–7040 (2011)
8. Shahzad, K., Elias, M., Johannesson, P.: Requirements for a Business Process Model Repository: A Stakeholders' Perspective. In: Abramowicz, W., Tolksdorf, R. (eds.) BIS 2010. LNBIP, vol. 47, pp. 158–170. Springer, Heidelberg (2010)
9. Elias, M., Johannesson, P.: A survey of process model reuse repositories. In: Dua, S., Gangopadhyay, A., Thulasiraman, P., Straccia, U., Shepherd, M., Stein, B. (eds.) ICISTM 2012. Communications in Computer and Information Science, vol. 285, pp. 64–76. Springer, Heidelberg (2012)
10. Elias, M., Shahzad, K., Johannesson, P.: A Business Process Metadata Model for a Process Model Repository. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Ukor, R. (eds.) BPMDS 2010 and EMMSAD 2010. Lecture Notes in Business Information Processing, vol. 50, pp. 287–300. Springer, Heidelberg (2010)
11. Porter, M.: Competitive Advantage. Free Press, New York (1985)
12. `http://jtc1sc32.org/doc/N0651-0700/32N0659.pdf` (last accessed March 10, 2012)
13. McCarthy, W.E.: The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment. The Accounting Review 57, 554–578 (1982)
14. Dunn, C., Cherrington, J.O., Hollander, A.: Enterprise Information Systems: A Pattern-Based Approach. McGraw-Hill/Irwin (2005)
15. Anthony, R.N., Dearden, J., Bedford, N.M.: Management Control Systems, Irwin Homewood, IL (1995)
16. Anthony, R.N.: Planning and Control Systems: A Framework for Analysis. Harvard Business School, Cambridge (1995)

17. Persson, A., Stirna, J.: Why Enterprise Modelling? An Explorative Study into Current Practice. In: Dittrich, K.R., Geppert, A., Norrie, M. (eds.) CAiSE 2001. LNCS, vol. 2068, pp. 465–468. Springer, Heidelberg (2001)
18. Moody, D.L.: The Method Evaluation Model: A Theoretical Model for Validating Information Systems Design Methods. In: European Conference on Information Systems, Naples, Italy (2003)
19. Davis, F.D., Bagozzi, R.P., Warshaw, P.R.: User acceptance of computer technology: a comparison of two theoretical models. Manage. Sci. 35, 982–1003 (1989)
20. Rescher, N.: Methodological Pragmatism: Systems-Theoretic Approach to the Theory of Knowledge. Blackwell Publishers, Oxford (1997)
21. Kandefer, M., Shapiro, S.: An F-Measure for Context-based Information Retrieval. In: International Symposium on Logical Formalization of Commonsense Reasoning, Toronto, CA, pp. 79–84 (2009)
22. Huang, A.H.: Effects of multimedia on document browsing and navigation: an exploratory empirical investigation. Information & Management 41, 189–198 (2003)
23. Mondary, T., Bouffier, A., Nazarenko, A.: Between browsing and search, a new model for navigating through large documents. HAL - CCSD (2007)
24. Melcher, J., Mendling, J., Reijers, H.A., Seese, D.: On Measuring the Understandability of Process Models. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) BPM 2009. LNBIP, vol. 43, pp. 465–476. Springer, Heidelberg (2010)
25. Zobel, J.: How reliable are the results of large-scale information retrieval experiments? In: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 307–314. ACM, Melbourne (1998)

Generating Event Logs with Workload-Dependent Speeds from Simulation Models*

Joyce Nakatumba, Michael Westergaard, and Wil M.P. van der Aalst

Eindhoven University of Technology, The Netherlands
{jnakatum,m.westergaard,w.m.p.v.d.aalst}@tue.nl

Abstract. Both simulation and process mining can be used to analyze operational business processes. Simulation is model-driven and very useful because different scenarios can be explored by changing the model's parameters. Process mining is driven by event data. This allows detailed analysis of the observed behavior showing actual bottlenecks, deviations, and other performance-related problems. Both techniques tend to focus on the control-flow and do not analyze resource behavior in a detailed manner. In this paper, we focus on *workload-dependent processing speeds* because of the well-known phenomenon that people perform best at a certain stress level. For example, the "Yerkes-Dodson Law of Arousal" states that people will take more time to execute an activity if there is little work to do. This paper shows how workload-dependent processing speeds can be incorporated in a simulation model and learned from event logs. We also show how event logs with workload-dependent behavior can be generated through simulation. Experiments show that it is crucial to incorporate such phenomena. Moreover, we advocate an amalgamation of simulation and process mining techniques to better understand, model, and improve real-life business processes.

1 Introduction

Process mining is a technique that extracts knowledge from *event logs* recorded by information systems [2]. Most systems are able to *sequentially* record *events* such that each event refers to an *activity* (i.e., a well-defined step in the process) and is related to a particular *case* (i.e., a process instance). The event logs of such systems also store more information about events, for example, the *resource* (i.e., person or device) executing or initiating the activity, and the *timestamp* of the event. Examples are Enterprise Resource Planning systems (SAP, Oracle), Business Process Management Systems (Hospital Information Systems) Using process mining techniques it is possible to discover processes from event logs. Moreover, event logs can be checked to assess conformance with respect

* This research is supported by the Technology Foundation STW, applied science division of NWO and the technology program of the Dutch Ministry of Economic Affairs.

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 383–397, 2012.

to defined processes and process models can be modified and extended using process mining techniques. This provides necessary insights to manage, control, and improve business processes [2]. In many of the business processes supported by information systems, human resources are the limiting factor, i.e., delays are often caused by the unavailability or overloading of people. Understanding such delays is vital for process improvement. Therefore, we focus on the analysis of event logs where most activities are executed by human resources.

Simulation is an important technique that can be used to, for example, analyze the performance of business processes, verify that a business process will work as designed, gain insights into the effects of particular business decisions. All of these tools use information about tasks, resources, and the ordering of tasks to calculate various performance indicators. Whereas simulation is a well established area of computing, there are several challenges when mapping a real life business process onto a simulation model especially when involving humans. Several pitfalls of current simulation models have been discussed in [1,3]. One of these is that the resource models built in simulations are usually very simple and are not a true reflection of reality. However, in this paper we deal with the modeling of resources in simulation models by adding additional parameters to the resource model in order to make sure that the process is modeled accurately. We focus on the problem of the *effect of workload on processing speeds*.

We provide a refined view on modeling resource behavior by taking into consideration the fact that the processing speeds of resources are not merely dependent on the service rates of the tasks but also on the workload present in the system. This is based on various studies in the area of psychology and operations research that suggest a relation between workload and performance of workers [5,13,16]. Such behavior is typical in organizations and in this paper, we build a simulation model based on colored Petri nets (CPNs) [6] that models workload influence on processing speeds of resources. Further on, we carry out experiments and the aim is to compare the simulated resource performance while taking workload into account with a simulated resource performance without the workload effect. Our experimental results indeed show that workload does have an effect on resource performance and should not be ignored when building simulation models. However, to adequately set the resource behavior parameters in simulation models, we can also exploit information available in event logs using process mining techniques. Given an event log, we can learn information characterizing workload-dependent speeds using process mining. This approach is implemented in the process mining framework ProM [14]. Moreover, we see the same relationship modeled in a simulation model also exists in real life based on the ProM analysis.

Accurate modeling of resource behavior in simulation models is important for two main reasons. First, a better modeling of resource behavior in simulation models will help make simulation models that are more realistic and also tightly coupled to Process-aware Information Systems [1,15]. Using simulations, we aim at the *generation and extension of event logs with workload-dependent speeds*. Log generation is a very important aspect because we are able to change model parameters or include different process alternatives in the simulation model, run simulations while obtaining new event logs. The generated event logs can be analyzed using process mining techniques to verify the effects of the parameter changes on resource behavior. This is not only relevant for analyzing alternatives for a concrete process, but also for the evaluation of process mining techniques focusing on the resource and time perspectives. Hence, event logs from the simulated environment and the real world can be compared using the same technique, i.e., process mining as shown in Fig. 1.

Second, the information about modeling resource behavior in simulation models can be used as a basis for *operational support and providing of more realistic simulations of resources*. For example, in [12] we discuss a testing platform for simulating resource behavior where users interacting with a workflow system are provided with on-line information about running processes and recommendations about the next actions to take in order to arrive at a goal [1]. Here, we provide for more realistic modeling of resource behavior in simulation models while taking workload present in the system into account.

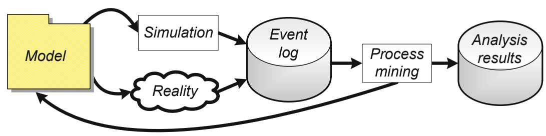


Fig. 1. Process mining is able to analyse event logs from different sources, i.e., Simulation and Reality

The remainder of the paper is organized as follows. First, we provide an overview of workload-dependent speeds in Section 2 and discuss the approach taken to model workload-dependent speeds in a simulation model. In Section 3, we discuss the CPN model developed for modeling the effect of workload on processing speeds. In Section 4, we discuss the experiments carried out based on the simulation model. Section 5 has a discussion of related work. Section 6 concludes the paper.

2 Background

In [1, 3], we discussed several problems that can arise when building simulation models involving human resources. For example, when resources are working they tend to distribute their attention over multiple processes, they also work part-time and in batches. However, when resources are working it is not only their distribution of attention over different processes that can affect their performance. In this paper, we argue that the absolute working speeds of resources also determine their capacity for a particular process.

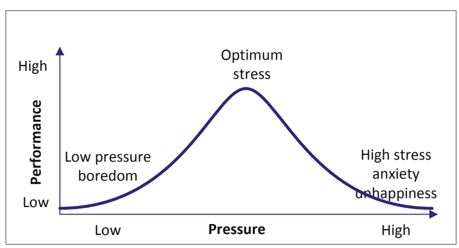


Fig. 2. Yerkes-Dodson Law modeled as U-shaped Curve

The speed at which resources work is in many systems partly determined by the amount of work that is currently present [13]. For example, in busy periods of the year people tend to increase their speed to process more cases. However, when people are given too much work over a long period of time, their performance then tends to drop. In the literature, this phenomenon is known as the "Yerkes-Dodson Law of Arousal" [16]. This law models human performance as an inverted U-shaped curve as depicted in Fig. 2. If the law holds, the performance of people (i.e., the speed at which they work) is determined by the workload that is currently present in the system [5, 11, 13]. This implies that for a given individual and a given set of tasks, there is an optimal level at which the performance of that individual has a maximal value and beyond this optimal level, the worker's performance collapses. In this paper, we do not try to represent this very complex world in a simulation model. However, we use simple parameters depending on the work-items present in the system. We limit our focus to a single process involving multiple resources and tasks and represent this in a simulation model.

2.1 Approach

When resources work, they execute a number of work-items for each case. In Fig. 3, we give an example of how work-items can be handled by two resources r_1 and r_2 over a 12-hour time period. The horizontal axis indicates the time over which work-items are executed while the vertical axis indicates the order in which the work-items are executed. Each box shown in the figure corresponds to a work-item executed by a resource. The start of the box indicates when a work-item is scheduled, the start of the shaded part of the box indicates when the work-item is started and the end of the box shows when the work-item is completed. For example, r_1 executes work-items w_1-w_9. and starts with the execution of w_1, followed by w_2 etc. From Fig. 3, we observe work-item executions at different points in time for each resource, i.e., point p_1 for r_1. If we consider point p_2 corresponding to resource r_2 we observe that:

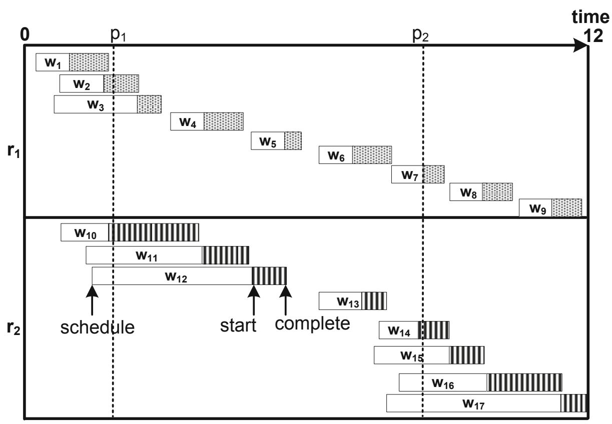


Fig. 3. Resources r_1 and r_2 execute work-items over a 12 hour period

- in the *allocated* (i.e., scheduled but not started) state, there are work-items w_{15}, w_{16} and w_{17},
- in the *executing* state, there is work-item w_{14}, and
- in the *finished* state, there are work-items w_{10}, w_{11}, w_{12} and w_{13}.

We have different properties of a work-item for example, when it was scheduled, started and completed, and we can use these to define further work-item properties that can be used to define workload.

Definition 1 (Work Item Properties). Let $\mathcal{R}$ be a set of resources, $\mathcal{T}$ be the time domain and $\mathcal{W}$ be a set of work-items. For any **work-item** $w \in \mathcal{W}$, we define the following **properties**:

- $t_{sch}(w) \in \mathcal{T}$ *as the schedule time of* w*,*
- $t_s(w) \in \mathcal{T}$ *as the start time of* w*,*
- $t_c(w) \in \mathcal{T}$ *as the completion time of* w*,*
- $r(w) \in \mathcal{R}$ *as the resource scheduled to execute* w*.*

From the three work-item states, we can define metrics for calculating workload. Although many workload definitions are possible [5,13], in this section we define workload based on the two main perspectives.

i. The *Queue Length* perspective specifies the amount of work scheduled for a given resource, and
ii. The *How Busy* perspective specifies the amount of work that each resource has executed in the recent past.

During the simulation it is possible to look at all the work-items that have been scheduled for a given resource before they start execution of a particular work-item. We now formally define workload based on the queue length perspective.

Definition 2 (Workload from Queue Length Perspective). Let $\mathcal{R}$, $\mathcal{T}$ and $\mathcal{W}$ be as defined in Def. 1. We define workload based on **queue length** perspective for a resource $r \in \mathcal{R}$ at a specific time $t \in \mathcal{T}$ by function $ql : \mathcal{R} \times \mathcal{T} \to \mathbf{N}$ where $\mathbf{N}$ is the set of natural numbers as: $ql(r, t) = |\{w \in \mathcal{W} \mid t_{sch}(w) \leq t \leq t_s(w) \wedge r(w) = r\}|$

Hence function ql counts the number of work-items that have been scheduled for a resource, however, at the point at which we measure workload, the work-items have been not been started. This can be computed on-the-fly by counting when a work-item is scheduled and this value can be decremented when the work-item is started.

We also define how to calculate workload based on the how busy perspective. However, in the how busy perspective we need to define a *horizon period* over which we can measure the number of completed work-items.We now formally define workload based on the queue length perspective.

Definition 3 (Workload from How Busy Perspective). Let $\mathcal{R}$, $\mathcal{T}$ and $\mathcal{W}$ be be as defined in Def. 1. Given a specific horizon period $h \in \mathcal{T}$, we define workload based on the **how busy** perspective for a resource $r \in \mathcal{R}$ at time $t \in \mathcal{T}$ by function $bu : \mathcal{R} \times \mathcal{T} \to \mathbf{N}$ as: $bu(r, t) = |\{w \in \mathcal{W} \mid t - h \leq t_c(w) \leq t \wedge r(w) = r\}|$

Here, we consider all the work-items that a resource has executed in a specific time period. Hence, given the current time t, we can define a time period for example, $t - h$ where h is a specific time period. If h is defined as 10 (cf. Fig. 3), this implies that we look at all the work-items that were completed no more than 10 time units before the resource starts execution of the current work-item. The schedule moment of a work-item can be derived as the point in time when a case arrives in the model. In this model, we are interested in the effect of workload on the processing speeds of resources. Although we here measure workload given a specific horizon period, it is also possible to use a function for assigning various weights to work-items depending on how long ago they were executed, assigning higher weight to items completed the same day but lower weight to items completed last week.

We characterize processing speeds as the flow time associated with each case. The flow time is the total time the case spends in the system, i.e., the difference between when the case is created and when the last task for the case is completed.

Definition 4 (Processing Speeds). Let $t_{sch}(w)$, $t_s(w)$, and $t_c(w)$, be as defined in Def. 1. We define the **service time** and **flow time** attached to each work item as:

- $st(w) = t_c(w)$ - $t_s(w)$ is the service time associated to w, and
- $ft(w) = t_c(w)$ - $t_{sch}(w)$ is the flow time associated to w.

The main parameters of the simulation model are as follows (a) *arrival rate* λ ($\lambda > 0$) the average number of cases arriving per time unit, (b) *service rate* μ ($\mu > 0$) the average number of cases that can be handled per time unit, and the *utilization* $\rho = \frac{\lambda}{\mu}$ is the expected fraction of time that the resource will be busy.

3 Modeling Workload Dependent Speeds in Terms of Colored Petri Nets

The effects of workload on processing speeds of resources were explored by simulating a CPN model. Colored Petri Nets (CPNs) is a high-level Petri net formalism that extends classical Petri nets with *data* (colored tokens), *time*, and *hierarchy* [8]. CPNs are bipartite directed graphs comprising of places and transitions and all the tokens in a particular place have a value of some common type. In CPN-terms, this means that all the tokens in a given place should belong to the same color set and each place has a color set (i.e., type). Tokens also have *timestamps* indicating when they can be consumed. When producing a token, it may be given a *delay*. This delay may be sampled from some probability distribution. Moreover, the time concept and the availability of many probability distributions in CPN Tools makes it possible to model performance aspects. By introducing resource tokens in the model, organizational aspects of the business process can be modeled. With the hierarchy concept of CPN Tools, it is possible to compose a CPN model in a modular way hence handling different levels of abstraction. CPNs can be distributed over so-called *pages* where one page can describe places and transitions and it may also refer to other pages [8]. Furthermore, CPN Tools have the ability to generate event logs which can be exported into ProM for analysis. Given the various functionality in CPN Tools necessary for modeling and simulating business processes, colored Petri nets were selected as the modeling language in this paper.

Our CPN model is a hierarchical model divided into six pages. The Overview (Fig. 4) page connects the Work Creation (Fig. 5) page and the Create Work Item page (Fig. 6). The CPN model handles tasks for a business process that deals with the managing of travel requests at some university. We do not discuss this process in detail here, however, our model is generic to handle any other process model. The process model consists of a number of tasks and for each task we specify a number resources that are allowed to it. Cases are generated in the Work Creation page based on a predefined distribution and are put in the State List together with their initial state. The time between two subsequent case arrivals is given by the function IAT() and the creation time of cases is recorded by the current model time function Mtime(). After a case has been added to the Queue it is sent to the Create Work Item page.

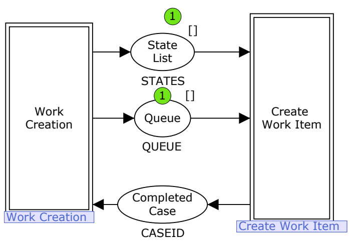


Fig. 4. The Main page

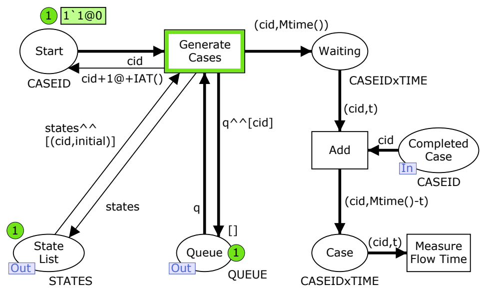


Fig. 5. The Work Creation page

Create Work Item. When the Queue arrives in the Create Work Item page in Fig. 6, Generate Random transition obtains the case that is at the top of the Queue and generates a random number used for making choices in the model where there are alternative paths have to be taken from a given place. The case together with the generated probability is added to the Case. Create Work Item transition takes a token from the Case and creates a new work-item. This transition obtains the next task to be executed for the case (using the State List and Process Declaration places), it also assigns a resource to execute the work-item (resources are obtained using predefined resource roles) and it also obtains the duration of the work-item. The work-item is added to the Work Item List which is sent to the Add Parameters page where resource parameters are calculated.

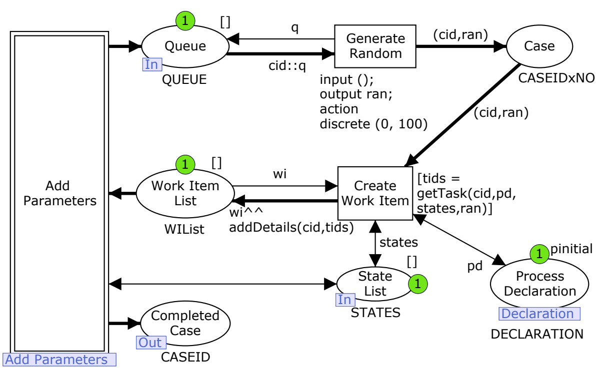


Fig. 6. The Create Work Item page

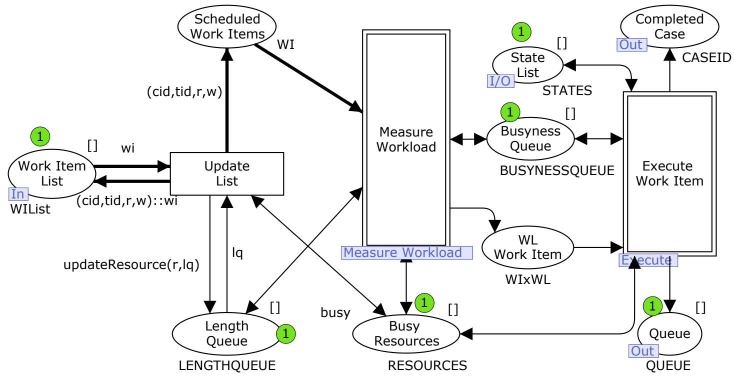


Fig. 7. The Add Parameters page

Add Parameters. When the Work Item List arrives in the Add Parameters shown in Fig. 7, Update List transition obtains each work-item from the Work Item List and adds it to the Scheduled WorkItems. Moreover, the Length Queue is also updated with information about the work-items that have been assigned to each resource. This is important because the information in the Length Queue will be used to determine the workload present in the model based on the *queue length* perspective. The Add Parameters page also connects the Measure Workload page (cf. Fig. 8) and the Execute Work Item page (cf. Fig. 9).

A work item is obtained from the Measure Workload page where its workload is measured and sent to the Execute Work Item page where it is actually executed. Moreover, if the work-item that has been completed in the Execute Work Item page is the last one for the case, then the case sent back to the Work Creation page through the Completed Case place.

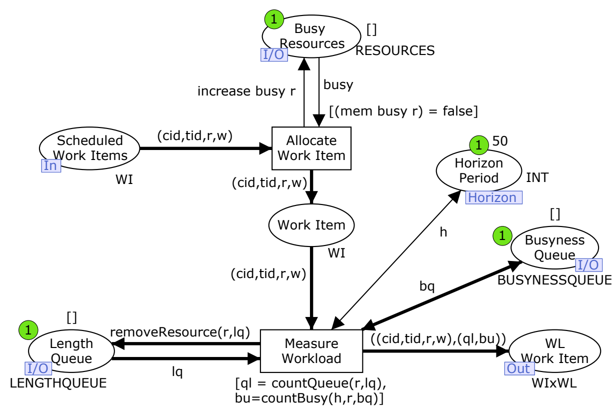


Fig. 8. The Measure Workload page

Measure Workload. In the Measure Workload page (Fig. 8), each work-item from the Scheduled Work Items is allocated to a resource to start its execution. This is only possible if the resource is not currently busy, i.e., is not in the Busy Resources. When Allocated Work Item transition fires, a work-item is taken from the list of Scheduled Work Items and added to the Work Item place. Moreover, the allocated resource will be added to the Busy Resources place. As soon as there is a token in the

Work Item place, Measure Workload will calculate the workload present before the allocated resource (r) starts execution. Measure Workload uses the information in the Length Queue and the Busyness Queues to calculate the workload present in the model.

– The Length Queue stores the number of work-items allocated to each resource. Therefore, function countQueue(r,lq) will return the number of work-items that have been scheduled for each resource (r) from the length queue (lq).
– For each resource, Busyness Queue will store a list for each that contains the the timestamps of work-items that each resource has completed so far. This information is used by the countBusy(h,r,bq) function to return the number of work-items from the busyness queue (bq) that have been completed in a specified period of time (h) for a resource (r). For example, if the given time period is $h = 100$, this function will returns the number of work-items completed during the previous 100 time units from the current time.

The measured workload information is added to work-item in the WL Work Item which is sent the Execute Work Item page. The number of work-items scheduled for the current resource are also reduced by 1 from the Length Queue.

Execute Work Item. The actual execution of a work-item is handled in the Execute Work Item page (Fig. 9). Here, the execution of the work-item is started and eventually completed. The duration of work-item processing is now dependent on the sampled duration from the Create Work Item page and the workload parameters from the Measure Workload page. We define a workload function $execTime(w, ql, bu)$ that takes the work-item duration (w), the queue length parameter (ql), and the how busy parameter (bu) to determine how long a resource will take executing a work-item. The function $execTime(w, ql, bu)$ uses the the power function to model the workload relationship and this implies with higher workload values the time of execution will be low. Moreover, the execution times will be higher if the amount of workload is low.

$$execTime(w, ql, bu) = w * 0.8^{((ql+bu)-1)} \quad (1)$$

Start Execution will move the work-item from the WL Work Item place to the Busy place. The length of work-item processing, i.e., done is given by the DurWL(w,ql,bu) function which is expressed in the CPN model but is equivalent to function shown in Equation 1. The current state for the case is updated in the State List based on the information in the pre and post sets from the process declaration. When Complete Execution transition fires, the resource is removed from Busy Resources and is now made available to execute scheduled work-items (if any). The State List is also updated with a new state for the case. The new state will have the number of tokens increased by 1 for the input place of task that will be executed next. If the task that has been completed is the last one for the case (given by finaltask), the case is sent back to the Work Creation page where its flow time is measured. Otherwise, it is added to the Queue and sent to the Measure Workload page where a new work-item for the case is created.

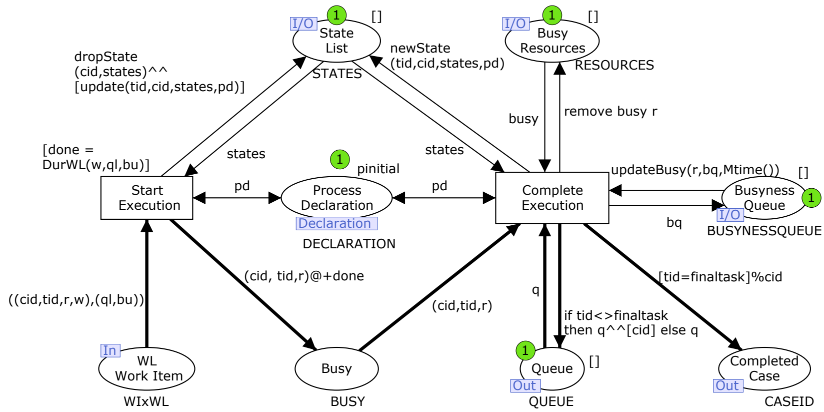


Fig. 9. The Execute Work Item page

The CPN model described creates an environment where we can analyze the impact of existing workload on the processing speeds of resources. In the next section we discuss experiments carried out to determine the effect of varying workload on the processing speeds of resources.

4 Simulation Experiments

Using the CPN model, experiments were carried out to investigate the effect of workload, i.e, denoted by the busyness and queue length perspectives on processing speeds, i.e., denoted by flow time. The monitor concept of CPN Tools allows measurement of performance indicators without changing the model [8] and were used to extract numerical data during the simulation experiments. The experimental results discussed here are based on simulations with 10 subruns, each subrun having 1,000 cases.

4.1 Experimental Results from Simulation Model

In this experiment, we compare the effect of varying arrival rates (λ) on the flow time values while keeping the service rates, i.e., $\mu = \frac{1}{15}$. The aim of this experiment is to compare the effect of increasing workload in the model on the flow time values and the utilization values. Considering the Yerkes-Dodson law shown in Fig. 2, we expect that the resource work slower when the workload is low but will eventually increase their speed as the workload increases in the model. Note that, although it is also possible for a busy resource to take longer while working, in our experiments we only focus on the notion that a busy resource works faster. This corresponds to the first half of the Yerkes-Dodson law.

We compare the flow time values obtained from two different input parameters that we use to determine the processing speeds of resources. In the first case, we consider experiments where we have assumed values for λ and μ. The processing speeds of resources is entirely based on μ (sampled from an exponential distribution). In this case the value of done shown in Fig. 9 depends entirely on the sampled service rate, i.e., w. We refer to the results from this experiment as NoWL.

For the second case, the processing speeds of resources are dependent on the workload present in the system. The value for done (used to determine the duration of execution) is calculated from the $execTime(w, ql, bu)$ function defined in Equation 1. This function takes into account workload present based on the queue length and the how busy perspectives. We refer to the results from this experiment as WithWL.

Experiment 1: Effect of Varying Arrival Rates on the Flow Time. In the first experiment, we measure the effect of varying arrival rates on the average flow time. The results are shown in Fig. 10(a). In the case NoWL, as the arrival rates increase, the flow times also increase. Typically, the flow times dramatically increase when ρ get close to 1. The second case, WithWL, as the arrival rates increase the flow time values also increase. However, if we compare the curves for the experiments from NoWL and WithWL, we see that when workload is taken into account, initially the flow time values are higher than the situation without workload. This is because resources work slower since the workload in the system is low based on the lower arrival rates of cases. Moreover, as seen in Fig. 10(a) when the arrival rate values increase the flow time values also increase and eventually the resource works much faster and this leads to lower flow time values compared to the situation where workload is not taken into account (here the flow times are low at the start of the experiment and increase much higher as ρ tends to 1).

Experiment 2: Effect of Varying Arrival Rates on the Utilization. The second experiment shows the results on the average utilization values based on varying arrival rates. The experimental results are shown in Fig. 10(b). For

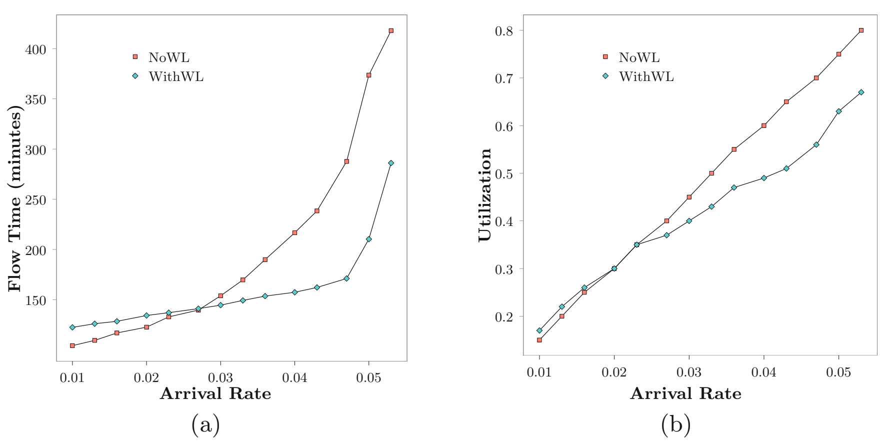


Fig. 10. The effect of varying arrival rates (λ) on (a) flow time and (b) utilization

comparison purposes, we obtained values for two scenarios NoWL and WithWL. Since the process model contains multiple resources, the values reported in this experiment are average utilization values. In the first case (NoWL), the utilization values follow a straight line because these are used as base line, for example, if $\mu = \frac{1}{15}$ and $\lambda = \frac{1}{18.75}$, then the expected utilization value is $\rho = 0.8$. However, if we compare the results from the second case with workload taken into account (WithWL), the utilization values are initially higher than the normal values because the resources work slower. However, as the arrival rates increase, yielding more workload in the system, the utilization values also increase but are lower than in the situation when resource work without workload effect on the processing speeds. This is because resources now work faster and hence will process more work-items. Therefore, the results from the experiments confirm that workload indeed has an effect on the processing speed of resources and eventually the utilization.

4.2 Analysis of Event Logs from a Real Life Process

This section is based on a real life case study taken from the CoSeLoG research project[1]. In the CoSeLoG project 10 Dutch municipalities are investigated as they execute their processes. The municipalities use workflow systems which generate event logs. The event log we use in this section is obtained from a process that deals with the handling of building permits. The log contains information about 324 cases, 19805 events, 15 resources and 175 activities. The start date of the log is "18-11-2009" and the end date is "12-01-2012". The results reported in this section are based on an extended implementation of a ProM plug-in described in [11] which quantifies the relationship between workload (using the queue length and how busy notions) and processing speeds (denoted by service and waiting times of events). Here, we use regression analysis to quantify the effect of workload (as the *independent variable* x),i.e., denoted based on queue length and how busy perspectives and processing speed (as the *dependent variable* y), i.e., denoted as service times of activities. The results are shown in Table 1 which include for each resource[2], the *correlation coefficient* (r) as the degree to which two variables are linearly related ($-1 \leq r \leq 1$) and the *r-square of the regression equation* (R^2, or the coefficient of determination), which is the proportion of variation in y accounted for by x [10].

The results in the table indeed confirm the effect of workload on the processing speeds of resources. Most of the R^2 shown in the table were higher than 0.7 which indicate that the variation in the service times is accounted for by the workload present in the system. For example, for resource "aweijter", "ckoets", "bwiling", their r and R^2 values were all high and this implies that as workload increases in the system, the speed of execution also increases. For, resources "cpers" in the 5th row of the table, their r and R^2 values were 0.97 and 0.94 respectively which implies a relationship between workload and the speed at which they worked.

[1] See http://www.win.tue.nl/coselog

[2] The resource names in Table 1 have been changed to ensure confidentiality.

Table 1. Linear regression results from real-life event log showing the relationship between workload and execution times of resources

resource	correlation coefficient	R^2	resource	correlation coefficient	R^2
aweijter	0.88	0.78	bwiling	1.00	1.00
ckoets	0.97	0.94	cokti	0.78	0.61
ejansen	0.92	0.85	spalm	0.9	0.81
phendric	1.00	1.00	brows	0.99	0.98
tjansen	0.99	0.98	cpers	0.76	0.58
mpauel	0.73	0.53	gursel	0.77	0.6

The results discussed in this section follow the first half of the Yerkes-Dodson law. Given that the simulation model generates event logs, it is now possible to analyse those logs using the process mining techniques that we have described in this section and in our earlier work [11].

Although varying workload has an effect on the speed at which resources work, when building simulation models this is rarely taken into account. From the experimental results discussed in this section, we see that workload has an impact on the flow times of cases, so if we assume that workload has no effect on the processing speeds, we may get performance results not corresponding to reality. Moreover, it is interesting that we not only show how to model such resource aspects in simulation models, but that such phenomena also exists in real life. Since, the event logs contains precise timestamps, we have been able to empirically investigate and quantify the effect of workload on processing speeds. Therefore, resources do not always work at constant speeds and there can be a number of factors that influence their speeds and in this section we shown that workload based queue length and the how busy perspectives indeed influences the resource processing speeds.

5 Related Work

The work presented in this paper is related to earlier work that has been done in the field of operations management and in simulation studies. The “Yerkes-Dodson Law of Arousal” [16] illustrated in Fig. 2, is one of the main motivations for this paper. Although the Yerkes-Dodson law originally related arousal to performance, this law has been extended to incorporate workload in the place of arousal. This law nominally depicts a low performance when the resource is over-worked or under-worked. In operations management, substantial work has been done to operationalize this “law” using mathematical models. Earlier work supporting this law shows that there exists a relationship between work-in-process and productivity [5,7]. The authors in [7] shows that the basis of industrial statistics collected that a strong correlation exists between productivity improvement and the speed of industry networks. In [13] queues with workload-dependent arrival and service rates are considered and the authors characterize the effect of work-in-process (the number of work orders on the shop floor) on productivity (the output per employee).

The discovery of simulation models is an approach that has been presented in [15]. Given an event log, a simulation model can be discovered and simulated using ProM and CPN Tools. In [9] a similar approach is presented that shows how to redesign a business process and predict its future performance based on simulation. The resource models used in both approaches are very simple, for example, the authors in [9] do not say anything about resources and their involvement in the simulation and redesign of business processes. However, to truly map a business process onto a simulation model, it is crucial that the resource perspective is modeled accurately. Although we focus on workload speeds in this paper, our earlier work also takes into account that people are involved in multiple processes, are available only part-time, and work in batches. Experiments show that these factors really influence performance [3].

The work in this paper is also related to earlier work presented in [11] where process mining techniques are used to analyse the relationship between workload and processing speeds based on event logs. In this paper, we focus on log generation from simulation models with workload dependent speeds.

6 Conclusion

Although organizations use simulation to analyze their business processes, the results may be very misleading if the assumptions used are incorrect. In this paper, we have addressed one of the problems discussed, i.e., human resources are often modeled incorrectly in simulation models. It is shown that resources are typically modeled in a naive manner and that this highly influences the simulation results. We have focussed in improving simulation models by providing a better modeling of resource behavior. The fact that people do not work at constant speeds and that workload has an effect on the processing speeds of resources has effects on the key performance indicators of a process. We characterized workload based on the queue length and how busy perspectives.

Based on a CPN model and with experiments, we show that the workload present in a system affects the processing speeds of resources. The information about the behavior of resources and also results from our earlier work [3,11] show that these characteristics should not be ignored when building simulation models. Moreover, simulation models can generate event logs contain information characterizing the workload relationship. This relationship is hidden, but correlations between workload and processing speeds can be observed using process mining techniques [11]. It is important to use both simulation and process mining techniques to better understand, model, and improve real-life business processes. Moreover, the relation can also be used as a basis for providing operational support where users are provided with recommendations about the next actions to take [4,12].

References

1. van der Aalst, W.M.P.: Business Process Simulation Revisited. In: Barjis, J. (ed.) EOMAS 2010. LNBIP, vol. 63, pp. 1–14. Springer, Heidelberg (2010)
2. van der Aalst, W.M.P.: Process Mining-Discovery, Conformance and Enhancement of Business Processes. Springer, Berlin (2011)
3. van der Aalst, W.M.P., Nakatumba, J., Rozinat, A., Russell, N.: Business Process Simulation. In: vom Brocke, J., Rosemann, M. (eds.) Handbook on Business Process Management 1: Introduction, Methods and Information Systems. International Handbooks on Information Systems, pp. 313–338. Springer (2010)
4. 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)
5. Bertrand, J.W.M., van Ooijen, H.P.G.: Workload Based Order Release and Productivity: A Missing Link. Production Planning and Control 13(7), 665–678 (2002)
6. CPN Tools, cpntools.org (2012)
7. Holström, J.: The relationship between speed and productivity in industrial networks: A study of industrial statistics. International Journal of Production Economics 34, 91–97 (1994)
8. Jensen, K., Kristensen, L.M.: Coloured Petri Nets: Modeling and Validation of Concurrent Systems. Springer, Berlin (2009)
9. Maruster, L., van Beest, R.T.P.: Redesigning Business Processes: A Methodology Based on Simulation and Process Mining Techniques. Knowledge Information Systems 21, 267–297 (2009)
10. Montgomery, D.C., Peck, E.A.: Introduction to Linear Regression Analysis. Wiley & Sons (1992)
11. Nakatumba, J., van der Aalst, W.M.P.: Analyzing Resource Behavior Using Process Mining. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) BPM 2009. LNBIP, vol. 43, pp. 69–80. Springer, Heidelberg (2010)
12. Nakatumba, J., Westergaard, M., van der Aalst, W.M.P.: An Infrastructure for Cost-Effective Testing of Operational Support Algorithms Based on Colored Petri Nets. In: Proceedings of the 33rd International Conference on Applications and Theory of Petri Nets. LNCS. Springer (2012)
13. van Ooijen, H.P.G., Bertrand, J.W.M.: The Effects of a Simple Arrival Rate Control Policy on Throughput and Work-in-progress in Production Systems with Workload Dependent Processing Rates. International Journal of Production Economics 85, 61–68 (2003)
14. ProM (2012), http://www.processmining.org
15. Rozinat, A., Mans, R.S., Song, M., van der Aalst, W.M.P.: Discovering Simulation Models. Information Systems 34(3), 305–327 (2009)
16. Wickens, C.D.: Engineering Psychology and Human Performance. Harper (1992)

FSM-Based Object-Oriented Organization Modeling and Simulation

Vojtěch Merunka[1,2]

[1] Czech University of Life Sciences Prague, Faculty of Economics and Management, Department of Information Engineering
[2] Czech Technical University in Prague, Faculty of Nuclear Sciences and Physical Engineering, Department of Software Engineering in Economy
vmerunka@gmail.com

Abstract. This paper presents the idea of the convergent approach to modeling and simulation of business requirements and software development based on the combination of the FSM and the Object-Oriented Approach. The first part of this paper discusses the motivation for the need to connect two areas of modeling and simulation: business requirements and software engineering. The second part of the paper presents the idea of modeling of processes and business situations as more mutually associated FSM representing particular objects and shows examples of this approach. The third and the last part of the paper presents the mapping of the proposed approach to BPMN-based and UML-based models and provides interesting new findings resulting from the proposed approach. This approach is based on our experience with our recent practical projects concerning business modeling and simulation in various application areas (e.g. health care, gas supply industry, regional management, administration process design of a new faculty of a university) and subsequent software development in these application areas.

Keywords: business process modeling, conceptual modeling, organizational modeling and simulation, object-oriented approach, FSM, BPMN, UML, BORM.

1 Introduction

Software application development for business and similar domain-specific areas shifts the attention at the requirement analysis and design activities, e. g. from the programming level to the modeling level. MDA (Model-Driven Architecture) [11] is the recent approach based on strategy of the application development based on requirements, conceptual and design modeling. The typical tool used in this area is the UML - Unified Modeling Language [19].

Our idea of continuous model-driven engineering aims to fill in the gap between of two worlds of "Business", which is process-based and requires deep management and economical knowledge, and "IT", which uses its own modern software development tools and techniques. This is to minimize the failure rate of information systems through the application of proper simulation

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 398–412, 2012.

and modeling techniques before the system is built. We want to advance the discipline of conceptual modeling in the area between the use of business domain knowledge and the use of modern advanced programming techniques and tools such as object-oriented programming environments (.NET, XCode, VisualWorks, ...), prototyping environments (Self, Squeak), non-traditional programming languages (Smalltalk, Objective-C) etc.

The goal of our paper is to converge the BPMN and UML modeling using approach, which will enable to use only one modeling and simulation paradigm trough the entire software system development life-cycle. Our paper contributes to the area of system modeling methodologies, tools and techniques to enable simulation, verification and validation activities.

2 Motivation

2.1 Our Experience

In our experience, any modeling and simulation tool and diagramming technique used at this kind of business projects should be comprehensible to the stakeholders, many of whom are not software engineering literate. Moreover, these diagrams must not deform or inadequately simplify requirement information. It is our experience that the correct mapping of the problem into the model and subsequent visualization and possible simulation is very hard task with standard diagramming techniques. We believe that the business community needs a simple yet expressive tool for process modeling; able to play an equivalent role to that played by Entity-Relation Diagrams, Data-Flows Diagrams or Flow-Charts over the past decades. One of the strengths of these diagrams was that they contained only a limited set of concepts (about 5) and were comprehensible by problem domain experts after few minutes of study. Unfortunately UML approach (as well as BPMN) lost this power of simplicity and clarity.

That is reason why we developed and successfully used our own BORM process diagraming technique [10] and our own way to start business system analysis. The initial work on BORM (Business-Object Relation Modeling) was carried out in 1993 under the support of the Czech Academic Link Programme (CZALP) of the British Council, as part of the VAPPIENS[1] research project; further development and recent practical projects in the last decade has been carried out with the support of Craft.CASE Ltd. - the British software consulting company supporting innovative technologies. (*VAPPIENS was funded by the British Governments CZALP, administered by the British Council. The authors acknowledge the support they received from this source, which enabled them to meet and carry out the initial work, out of which BORM grew.*) BORM has been used in last 15 years for a number of business consulting and software engineering projects including

- the identification of business processes in metropolitan hospital,
- the modeling of properties necessary for the general agricultural commodities wholesale sector requested by the Agrarian Chamber,

[1] Visual Application Programming Paradigms for Integrated ENvironmentS.

- as a tool for business process reengineering in the electricity supply and gas supply industry (see example in figure 1),
- as a tool for business process reengineering for telecommunication network management,
- in organizational modeling and simulation of regional management project concerning the analysis of the legislation and local officials' knowledge such as living situations, law, urban planning etc. (see simulation example in figure 2).
- several business process simulation projects in area of simulation of marketing chains for Makro,
- visualization of safety and fire regulations in the electric power engineering sector and
- administration process design of a new Faculty of Information Technology of the Czech Technical University in Prague.

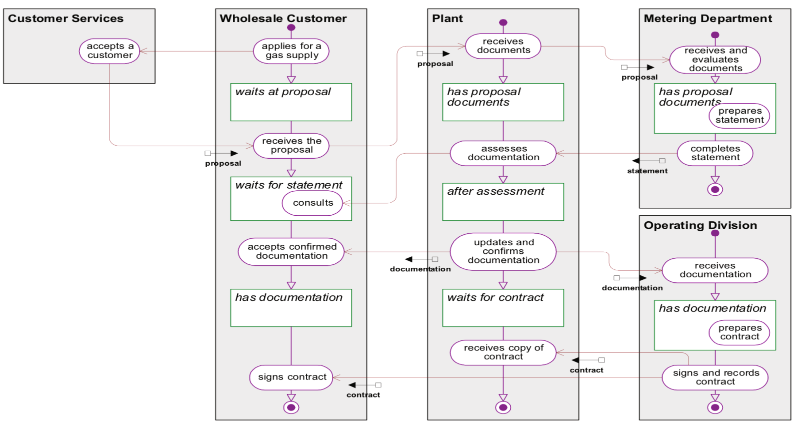


Fig. 1. Workflow process of contract on gas supply

However during the last decade there was an significant upgrade of UML, and also a new standard for business process modeling BPMN has been developed and we recognized that our approach is close to both of them. We think that based on our past experience, we can propose a new approach. This new approach is based on the object-oriented concepts, is using the well proven technology of FSM for modeling and simulation. It has almost the same expressive power as the UML and the BPMN together, but it is expressed in a uniform and simpler manner.

2.2 Motivation - The Gap between the World of Business Modeling and Software Modeling

Nowadays, there is a great variety of tools and techniques for business modeling. Unfortunately, there is no standard for business modeling like UML for software

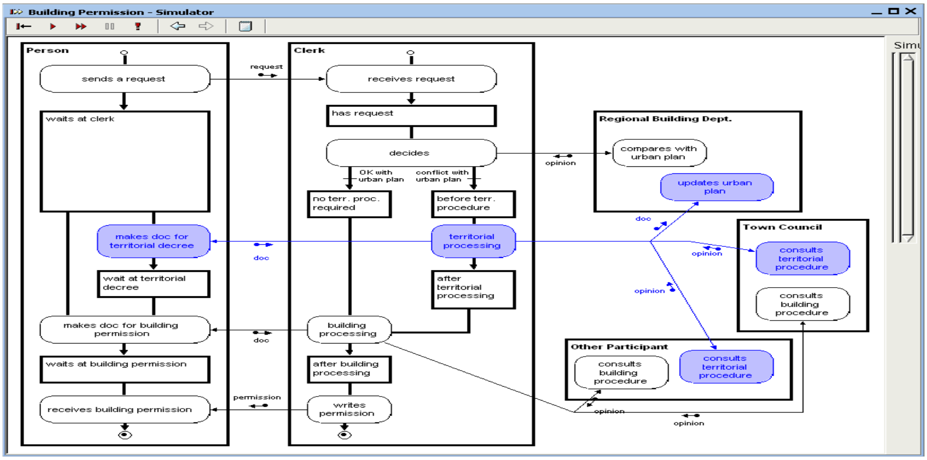


Fig. 2. Simulation of the regional management process on building permission

engineering yet. Nevertheless, we can presume that the approach to become a standard will be the BPMN. [2,17,7] In Europe, Aris and its EPC diagram is still very popular; however BPMN authors say this technique becomes obsolete and is almost unknown outside the Europe.

When OOP started to be used in practice, it was assumed that object-oriented technology will become mainstream in software development (which was correct) but also that object-oriented approach will affect the approach to business process modeling, organizational engineering and the whole area of activities preceding the formulation of information system requirement. That, unfortunately, did not happen. These pre-implementation activities are still carried out the old fashioned way, which results in a semantic gap between the world of business modeling and the world of software modeling as shown in figure 3. Latest publications even express opinions such as "OOP has failed, OOP is a dead end, etc.". These are written by people who have no experience with pure object-oriented languages and environments, but only with hybrid ones (e.g. Delphi, Java, etc.) and they generalize their negative practical experience with failed projects to the whole paradigm.

Expected output of the business engineering activities is information or data in a form that can be directly used as an input for implementation of the system in the spirit of software engineering. However, this is not the easy case; there are following issues described by Illgen and Hulin in [9] and Van der Aalst in [1]:

1. *oversimplification* - while trying to at least finish business and organizational model we are forced simplify the problem being modeled and
2. *inability* - some important details cannot be recorded because of the method being used.

We believe together with Schach [16] that this is the reason why software system modeling and subsequent design of business applications is a pretty hard deal: Today's world of software development still works with algorithmic and imperative style of thinking, and therefore the produced models concentrates on functions, function separation, continuity etc. But this "behavioral" and "straight timeline" approach goes against the business modeling paradigm which mainly focuses on states, situations, rules - often even in a form of statutory regulations and directives and concurrency. Behaviors are of less importance and make sense only if we know what they are good for. That is why it is much more natural to start analyzing a business process from the description of initial situation (e.g. we have goods, we want to sell it, the customer wants our goods and has money, ...) and the description of the desired final situation (e.g., the paying customer makes us a good profit). In this perspective, the sequence of behaviors is only a result of an effort to get from the initial to the final situation, thus, being much more declarative then imperative style of modeling. The significance of the declarative approach is expected to grow even in the software programming itself, where the imperative (e.g. behavior-oriented) modeling style is still prevalent, although unnatural for business and organizational modeling.

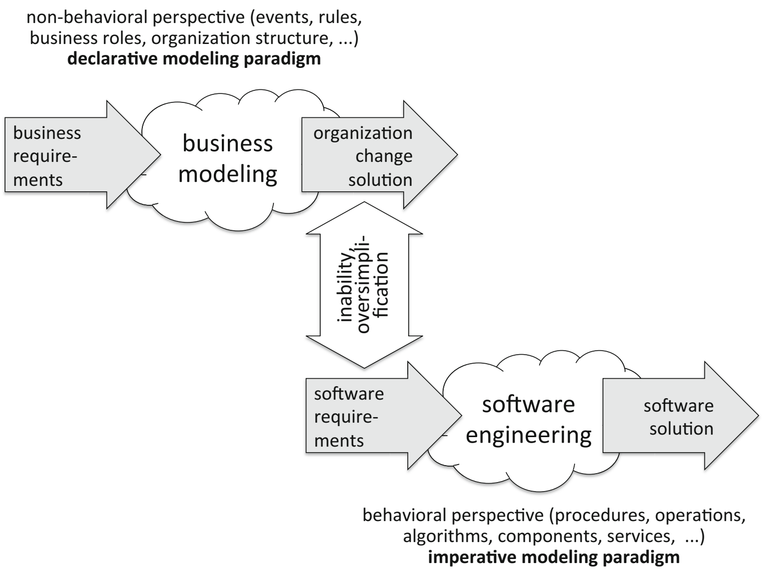


Fig. 3. Semantic gap

2.3 Gap between Software Modeler and Software Programmer

This second gap persisting from 1970's is known as the impedance problem. It concerns the different interpretation of conceptual relations by the analysts and by the programmers. This gap is caused by different interpretations of ontological categories of the supertype-subtype relationship (inheritance), whole-part relationship (composition) and association relationship by various kinds of programmers and developers. This problem has significant influence to the modern application development in case of so called object-relational impedance mismatch (mentioned by Hohenstein in [8], for example), where relational databases on the server side stress association relationship (in fact the only native relationship supported by the relational data model), but client-side applications are based on the object-oriented programming languages, in which we can work with the inheritance and compositions, but we don't have any direct language support for associations (see table 1). This leads to the need to permanently transform data from their server data model to their client model and vice versa. In this situation, it is pretty hard for analysts to maintain the correct usage of associations and compositions in models they need to deal with, as the essential difference between these two relationship concepts is usually eclipsed by their implementation in a programming environment.

Table 1. Impedance problem

	object-oriented programming languages (client side)	relational databases (server side)
inheritance (is-a relationship)	YES	NO
compositions (whole-part relationship)	YES	NO
associations	NO, must be implemented via compositions	YES

3 Our Solution - FSM and OOP Combination

Our solution is based on the reuse of old thoughts from the beginning of 1990s regarding the description of object properties and behavior using finite state machines (FSM). The first work expressing the possible merge of OOP (Object-Oriented Paradigm) and FSM was the Shaler's and Mellor's book [14]. One of the best books speaking about the applicability of OOP to the business modeling was written by Taylor [18]. These works together with our practical experience is why we believe that the business requirement modeling and simulation and software modeling could be unified on the platform of OOP and FSM.

Figure 4 shows and example of a model of a book in a library represented in a form of a finite state machine with three states: a book on a shelf, a book on loan and a returned book to be put back on a shelf. These states are easily recognizable through an interview with domain experts. When these states are recognized, it is possible to identify behaviors required for transferring books between the states. Of course, other objects in the system can be modeled this way too. Let us take a borrower as an example. Even him has his own states which can be related to the states of the book; and analogically, the borrower's behaviors can be related to the book's behaviors, see figure 5. This figure 5 shows that a satisfied borrower is associated with an object of a book on loan in

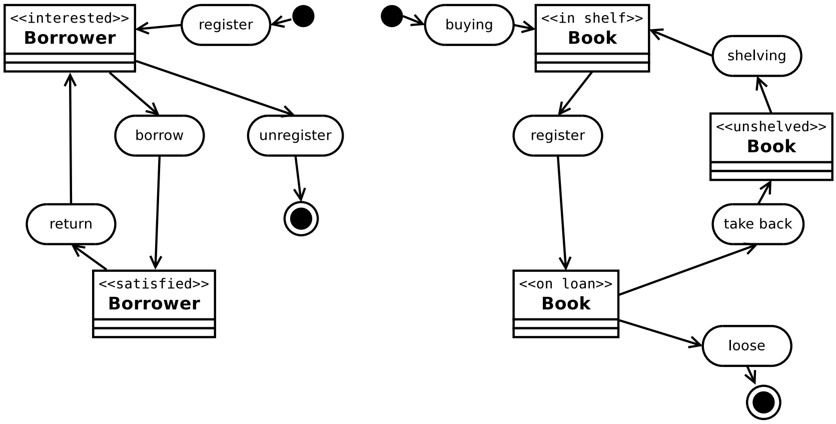


Fig. 4. A book and a borrower as FSM

a relationship with cardinality of a 1 to many. That means, that every book on loan has to have its borrower and every satisfied borrower is satisfied if and only if he has at least one book borrowed. This is the possible business situation of our model. Similarly, we can identify the associations between borrower interested in borrowing a book and available books in the shelves. This association can be another possible business situation of our model.

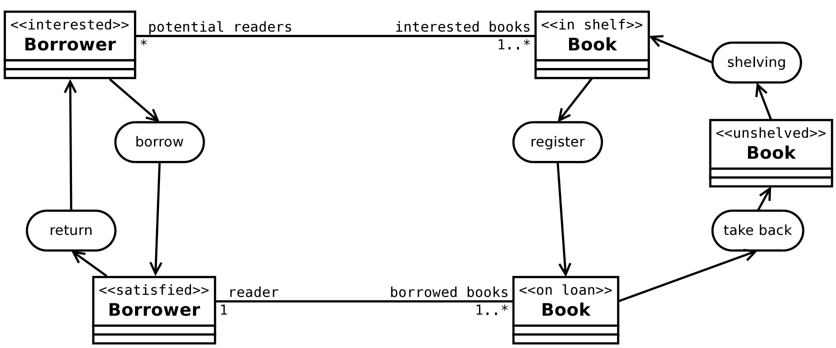


Fig. 5. Relationships (associations) between a book and a borrower

Moreover, the notion of association between states of objects can interestingly be generalized for the links between behaviors. It is the link between the activity borrowing and activity registration in the figure 6. This new kind of relationship is analogous with data association and can be regarded as the generalized message passing already existing in dynamic UML models.

This model can be further processed. One option is to further refine or conversely simplify (aggregate) the detail of the model. Figure 7 gives an example of

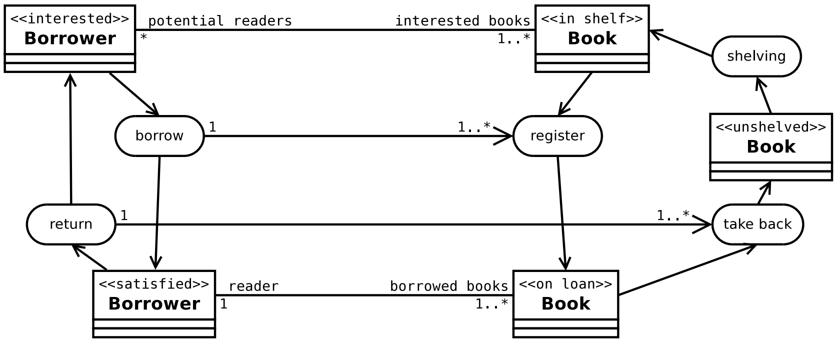


Fig. 6. Communication links between activities of a borrower and activities of a book

two levels of detail of the same model. The lower model is the same model as in figure 5, but the one at the top is its simplified - aggregated - and very familiar form. It is a common model of a data structure, part of the UML class diagram, which shows a borrower and a book connected with a relationship-association (* to 1..*). This figure also illustrates that this relationship can be built as a union of two more detailed situations from the model at the bottom. For this act of refinement or aggregation we can conveniently use the well known conceptual relationship of supertype-subtype or whole-part (composition).

Besides changing the level of detail of models, we can also filter these object-oriented FSM models. Filtering is an operation of simply removing an unnecessary detail. In the figure 8 there is a model with removed state returned book and related behaviors, because the borrower has not any direct connection to these parts of the model and therefore they are unnecessary from his point of view.

It is obvious that the operations of filtration and simplification-specification can be combined. In the figure 9, the borrower is simplified into a single object class with no states and a book has only two states being important for borrowers.

4 Discussion

The presented modeling approach unifies UML-style object modeling and business process modeling in the business and organizational engineering techniques and tools style. Models like UML, BPMN and other can be easily derived from this model. Overall schema of this unified modeling approach is laid out in the figure 10.

From the viewpoint of this schema, we can consolidate the terms used in modeling with BPMN and UML into a table 2.

The proposed unified approach generalizes object modeling. In this perspective, modeling in UML and BPMN could be perceived as mutually following stages of the new more universal approach according to the MDA principles

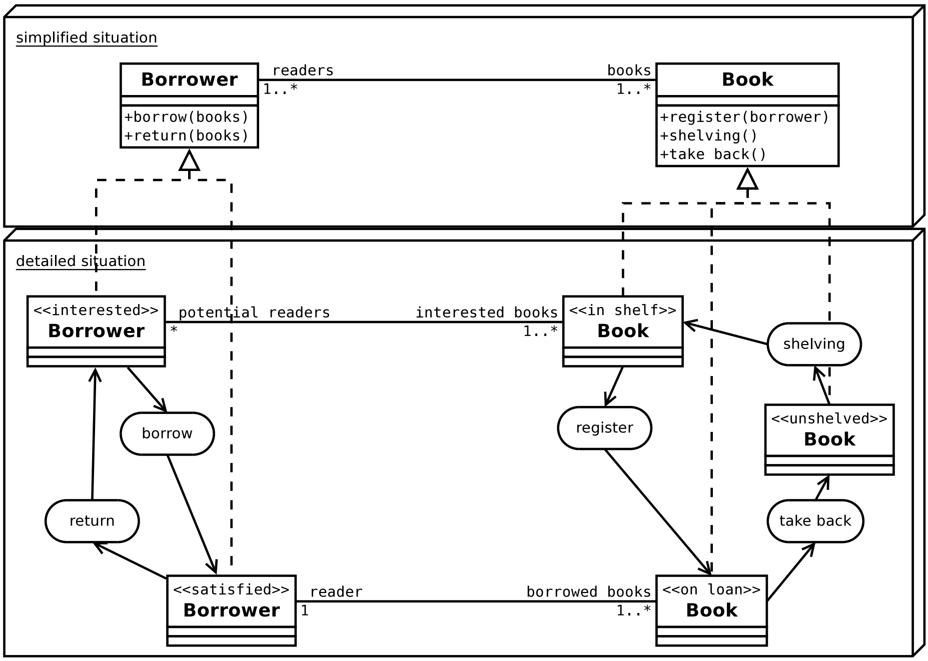


Fig. 7. Simplified and detailed view of the same model

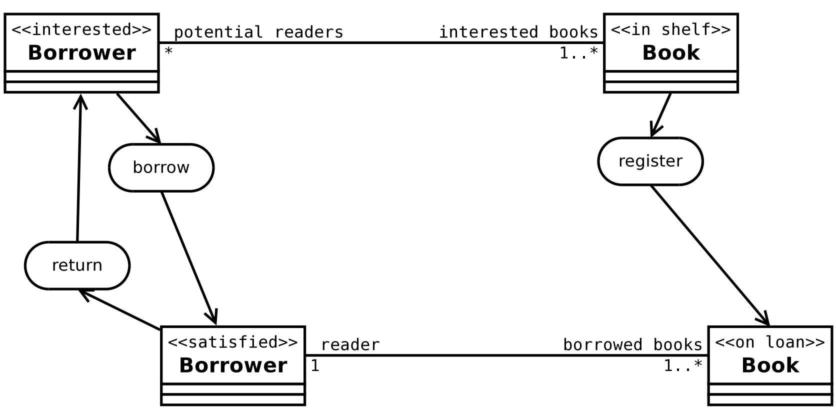


Fig. 8. Filtered model

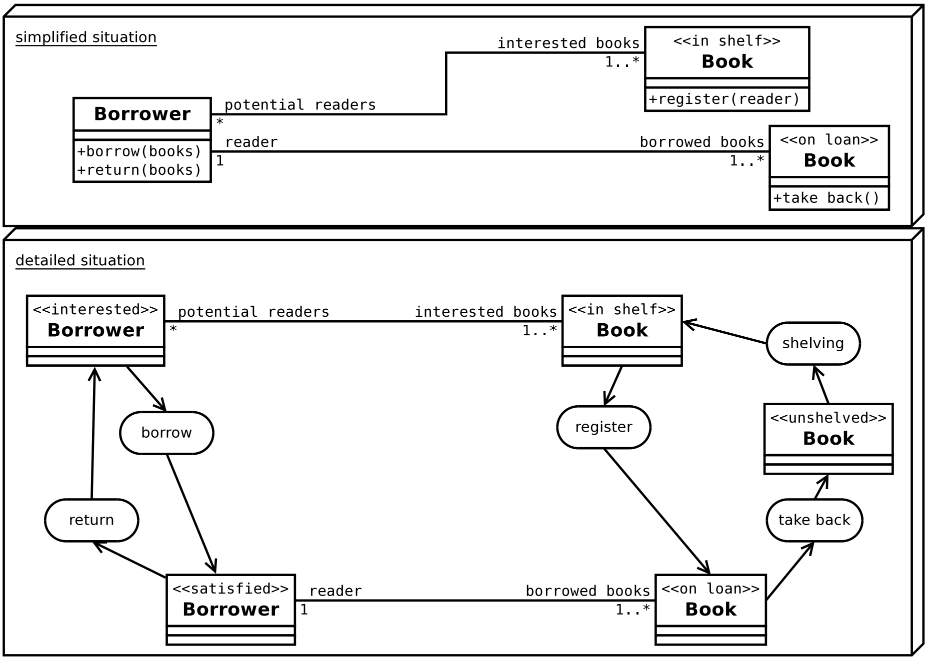


Fig. 9. Simplified model using both filtration and generalization

[11]. Figure 11 shows mapping or our model to the standard BPMN and figure 12 shows the same model expressed in the state-chart of the standard UML. Possible advantages of our convergency approach follows:

1. BPMN and UML both cover only a *subset* of the entire exploitable space of modeling concepts (see table 2).
2. The most important concepts are *states* of objects. Behaviors represent only the necessary glue between them. Both business processes and software components should be therefore modeled by starting with their states - *situations* of participating objects in the requested structure in some time. Modeling can be easier, more precise and less behavioral-imperative then it is today.
3. Our approach also denotes interesting *conceptual difference between compositions and associations*. The composition represents a greater detail of the same object, but the association is an information relationship between two different objects. This differentiation is very valuable especially because programmers often mix up these two terms. From a certain point of view, the composition can be seen similar to IS-A relation (in order to express detail of the concrete object) rather than to association which is really about information relationship between different objects. Mode detail about these conceptual differences is described in [13].
4. A new modeling term was defined - *communication* - which de facto generalizes the OOP term message. Communication is a link between behaviors

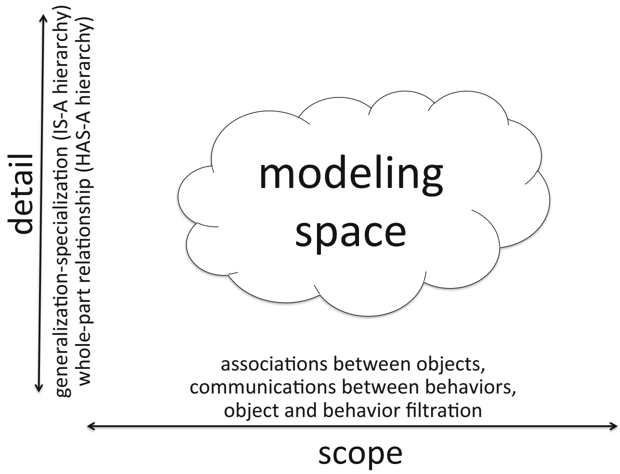


Fig. 10. Modeling space

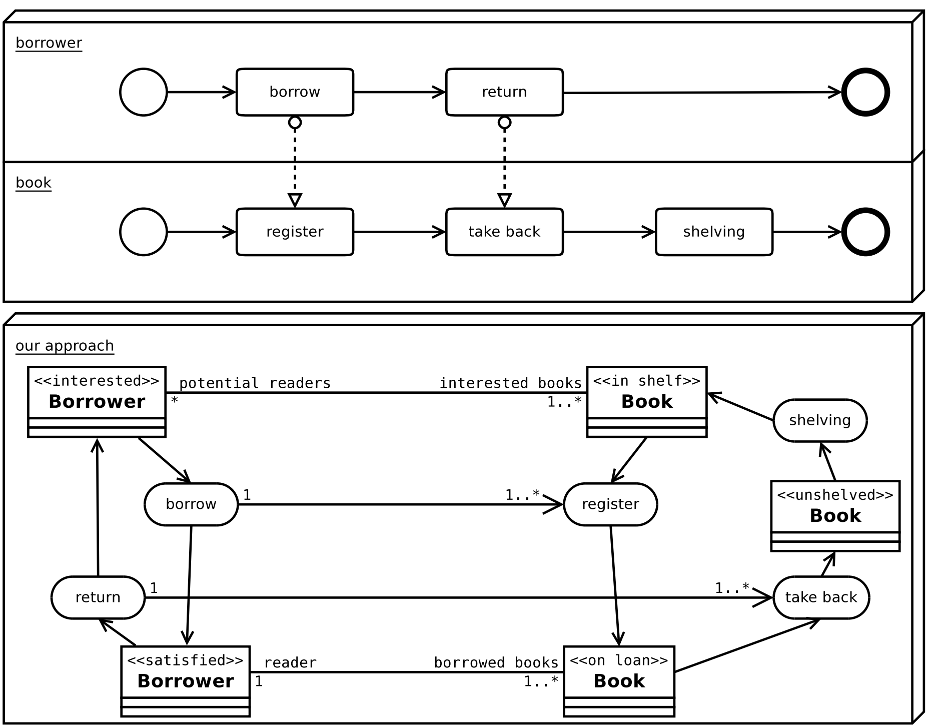


Fig. 11. Mapping to BPMN

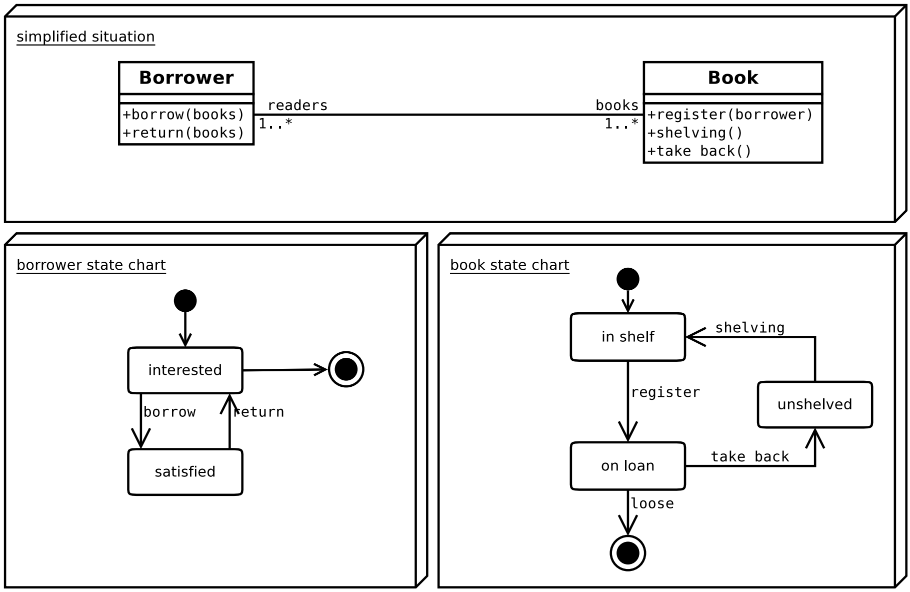


Fig. 12. Mapping to UML

Table 2. Modeling Concepts Coverage

our approach	BPMN-based approach	UML-based approach
objects	swimlines only (process participants)	YES
object states	can be expressed by various events	YES
associations between objects (data links)	NO	YES
associations between object states (data links)	YES	NO
object behaviors (activities)	YES	YES
communications between behaviors (messages)	YES	YES
generalization-specialization relationship (inheritance)	NO	objects only
whole-part relationship (composition)	NO	objects only

of different objects whose states could be also associated. This term communication is symmetric to the term association and can have own cardinality as well (although we would recommend using the term “frequency”). Association is an information relationship between objects or their states, whereas communication is an analogous relationship between the behaviors of these objects.

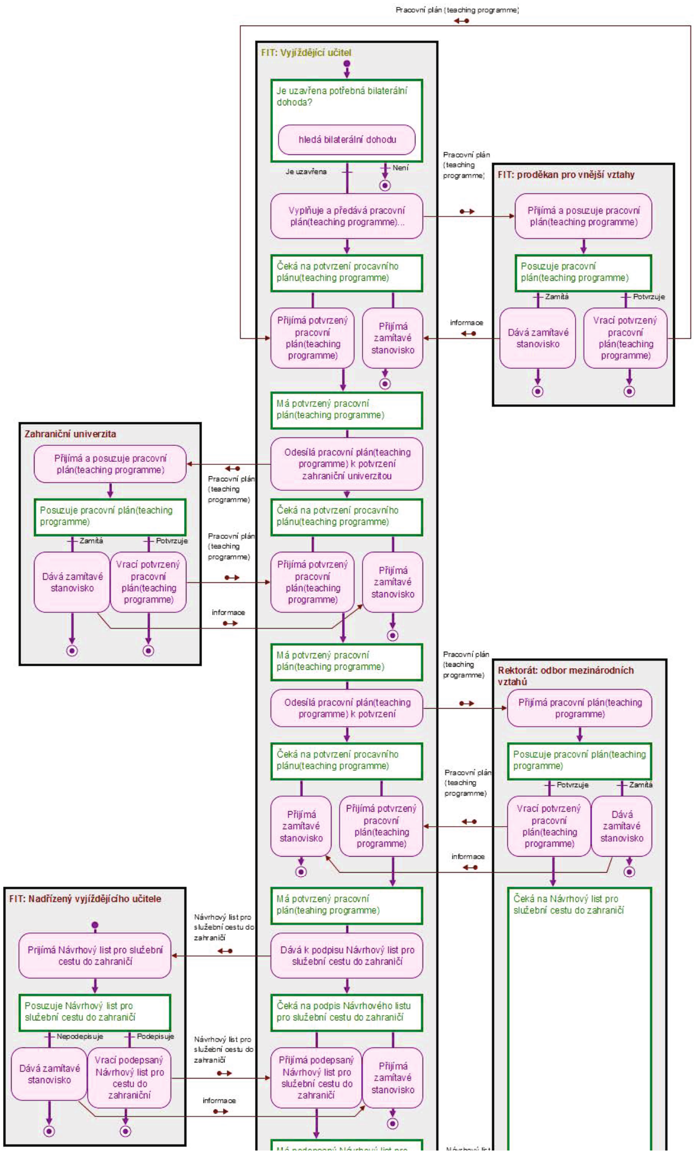


Fig. 13. ERASMUS Student Mobility for Studies - a fragment of a real simulation model

This strong difference between the concept of association (e.g. the relationship between different objects on the same level) and composition (e.g. the tool for making more or less detail of some particular object) can help the software developers to solve the impedance problems.

Figure 13 brings an example of a real process. This is the fragment of the process ERASMUS Student Mobility for Studies made for the new Faculty of Information Techology of the CTU Prague. There was total number of 63 similar business processes and 55 participants (objects), where each of them has been modelled as a FSM and connected in various ways with other FSM. Average number of interconnected FSM in one process was 7 and the total number of micellaneous data flows (e.g. documents, forms, e-data, ...) running between particular FSMs was 72.

5 Conclusion

In this paper we presented the idea of the convergent approach to modeling and simulation of business requirements and software development. Our approach called BORM is combining the object-oriented approach and finite-state machines and is based on our practical experience with recent project, which were aimed to help the teams made by business consultants and software developers from various areas (e.g. health care, gas supply industry, regional management, university management). We feel that the highest value of our approach is generated by the way of modeling, which smoothly connects two different worlds: business engineering and software engineering. We believe that this approach can help in future possible integration of BPMN and UML models for complex projects requiring the strong collaboration between software system architects and problem domain experts in area of organization structures modeling and subsequent simulation as it is predicted by Scheldbauer in [15].

Our future work will be focused on implementation of the proposed concepts in selected CASE tools and on incorporation of this approach into the BORM methodology [10]. The hot candidate for this project is the MetaEdit CASE tool by the Finnish company Metacase Ltd. [12] The autor would like to acknowledge the support of the research grant SGS11/166/OHK4/3T/14 and NAKI MK-S-3421/2011 OVV.

References

1. van der Aalst, W.: Business Process Simulation Revisited. In: Keynote Speech at the EOMAS Workshop (2010), http://www.eomas.org (cit: April 10, 2011)
2. Allweyer, T.: BPMN 2.0. Books on Demand GmbH, Norderstedt (2010) ISBN 978-3-8391-4985-0
3. Barjis, J.: Developing Executable Models of Business Systems. In: Proceedings of the ICEIS - International Conference on Enterprise Information Systems, pp. 5–13. INSTICC Press (2007)
4. Barjis, J., Reichgelt, H.: A Petri Net Based Methodology for Business Process Modeling and Simulation. In: The Proceedings of the Fourth International Workshop on Modeling, Simulation, Verification and Validation of Enterprise Information Systems (MSVVEIS), Paphos Cyprus (2006) ISBN 972-8865-49-X
5. Degen, W., Heller, B., Herre, H., Smith, B.: GOL - Towards an axiomatized upper level ontology. In: Proceedings of FOIS 2001, Ogunquit, Maine, USA. ACM Press (2001)

6. Eriksson, H., Penker, M.: Business Modeling with UML. John Wiley and Sons (2000) ISBN 0-471-29551-5
7. Grosskopf, A., Decker, G., Weske, M.: Business Process Modeling Using BPMN. Meghan Kiffer Press (2006) ISBN 978-0-929652-26-9
8. Hohenstein, U.: Bridging the gap between C++ and relational databases. In: Cointe, P. (ed.) ECOOP 1996. LNCS, vol. 1098, pp. 398–420. Springer, Heidelberg (1996)
9. Ilgen, D., Hulin, C.L.: Computational Modeling of Behavior in Organizations - The Third Scientific Discipline. American Psychological Association, Washington, DC (2000) ISBN 1-55798-639-8
10. Knott, R.P., Merunka, V., Polak, J.: The BORM methodology: a third-generation fully object-oriented methodology. In: Knowledge-Based Systems. Elsevier Science International, New York (2003) ISSN 0950-7051
11. MDA – The Model Driven Architecture, OMG – The Object Management Group, http://www.omg.org
12. MetaCase - Domain-Specific Modeling with MetaEdit+, http://www.metacase.com
13. Molhanec, M.: Conceptual Normalisation Formalised. In: Barjis, J., Eldabi, T., Gupta, A. (eds.) EOMAS 2011. LNBIP, vol. 88, pp. 159–172. Springer, Heidelberg (2011)
14. Shlaer, S., Mellor, S.: Object Lifecycles: Modeling the World in States. Yourdon Press (1992) ISBN 0136299407
15. Scheldbauer, M.: The Art of Business Process Modeling–The business Analyst Guide to Process Modeling with UML and BPMN. Cartris Group, Sudbury MA (2010) ISBN 1-450-54166-6
16. Schach, S.: Object-Oriented Software Engineering. McGraw Hill, Singapore (2008) ISBN 978-007-125941-5
17. Silver, B.: BPMN Method & Style. Cody-Cassidy Press, Aptos (2009) ISBN 978-0-9823681-0-7
18. Taylor, D.A.: Business Engineering with Object Technology. John Wiley (1995) ISBN 0-471-04521-7
19. The UML standard, OMG – The Object Management Group, ISO/IEC 19501, http://www.omg.org

Enterprise Systems Meet Social BPM

Martin Molhanec

Czech Technical University in Prague
Faculty of Electrical Engineering, Department of e-Technology
molhanec@fel.cvut.cz

Abstract. This article engages in issues of Enterprise Systems and SBPM (Social Business Process Management) influence, integration, joint evolution and reasons which cause these matters of fact. Afterwards, the Author describes his own approach based on extensions of WebRatio CASE tool, primarily targeted as a tool for supporting WebML method, to assist in modelling, developing and operating enterprise SBPM systems. Subsequently, the Author proposes several design components intended for BPM developers which can be easily used to create a SBPM system with features as a collaboration of users, dynamic changes of workflow in runtime, common cooperation through Facebook or Twitter, etc. In the end, the Author summarizes the paper, suggests questions and discusses future directions of implementation of Enterprise SBPM Systems, especially from the point of view of an employment of WebRatio CASE tool as an analytic and design tool as well as operating platform into one.

Keywords: Enterprise systems, BPM, BPMN, social networks, WebRatio, WebML, social features, web modelling, software engineering.

1 Motivation

Modern WOIS (Web Oriented Information Systems) strongly targeted for use inside a given enterprise or business domain represent a modern information operating platform for enterprise business processes running in it. For modelling of such complex WOIS there are several web oriented analytic methods, such as OOHDM, UWE, and the most develop method of them – WebML [1 and 2]. On the other hand, in the area of business systems the specialised business process modelling methods are used, such as BPMN [3], BORM [4] and Ontology [5]. It is evident that the ones analysing and modelling methods exist in the area of web engineering and the other methods exist in the area of business modelling. We may put down a question if there is any connection between these two types of analysing methods as well as if exist the methods and tools supporting both such different analysing and modelling approaches ever.

The last decade moved business into a very dynamic environment which is very liable to changes caused by global market situation, intensive competition and continuing dissemination of new technologies. Such changes must be supported not

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 413–424, 2012.

only by organisational structure, company workflow and business processes but also by information systems themselves.

In addition, a modern information system is highly complicated application. A complete analysis of that is traditionally required as a necessary part of first phase of software construction process. Nevertheless, not every time it is possible to make a final analysis as deeply as needed. As time goes on requirement changes constantly coming and such changes aren't often caught by analysis.

In recent years there has been an enormous increasing of Business Process Management (BPM) usage [6, 7, and 8]. One of the ways how to catch changing requirements is to use social BPM. It involves ordinary users to catch the changes and requirements. But it is not quite clear at present time what meaning the idea of social BPM is [9]. We will discuss this subject in more detail thereinafter.

We can summarize the subject so far as follows. Present-day BPMs are not just about modelling process, but no less, about collaborating of different users as well. Interested users need to communicate and cooperate during the course of model evolution. Finally, these users aren't usually skilled in business processes modelling subject at all in contrast to a present-day trend to involve nontechnical users to take a part in a development process too.

The rest of this paper is organized as follows. Next, in Section 2, we start off by defining the problem statement we must resolve. Section 3 describes WebML method and WebRatio CASE Tools, Section 4 explains connection between WebRatio CASE tool and BPMN notation, finally section 5 introduces our approach. After that, Section 6 gives the results of our work, i.e., our implementation of Social Features in WebRatio CASE tool, and finally, Section 7 summarizes the paper, suggests improvements and discusses future directions for research.

2 Problem Statement

As Clay Richardson says, the Social BPM is: "*Processes developed and improved through the use of social technologies and techniques*". This definition follow his personal view of it as a way of bringing a new wave of BPM suites, including process Wikis, Process Mashups and BPM-as-a-Service started a lot of activity on the web [9]. But there are other opinions of this subject, under mentioned, that have been collected from social BOM discussion on EbizQ [10].

According to Michael zur Muehlen: "*Social is all about providing context, a rich environment of data points that a streamlined workflow would be lacking otherwise. The challenge is to make this context useful, both from a social networking perspective and from an unstructured data perspective*", [9]. But, Tom Allanson explains that: "*Social BPM is basically just collaborative business process management utilizing a collective network environment - it's about extending BPM access and decision-making to partners and select external parties without compromising the exclusivity of the core group*", [Ibid.].

It can be seen that there is not only one right opinion about this subject. By the Authors, there are two large areas, in which the feature of social BPM can be implemented.

- The first one is a collaboration platform for developers, business analysts and other professionals around BPM.
- The second area is a collaboration platform for users of BPM systems.

In this paper we aim only at the second area of implementing social BPM features. There are several issues we must solve. One issue is how to make BPM modelling dynamic (so the model can change during running the system). Another issue is how to easily involve non-technical users so they can change the model, as it evolves in the time, without a special business modelling knowledge.

3 WebML and WebRatio

WebML [2] is the most developed web-method at the present time. As stated in the Wikipedia [1].

> *"WebML (Web Modeling Language) is a visual notation for designing complex data-intensive Web applications. It provides graphical, yet formal, specifications, embodied in a complete design process, which can be assisted by visual design tools, like WebRatio. This method has five models: structure, derivation, composition, navigation and presentation. These models are developed in an iterative process."*

At first sight it looks like a process model is not included in the WebML concept or if you like, that any of the five different diagrams do not deals with the modelling of process domain of information systems. But, it is not true at all. The model of processes is included into a composite model of composition model and navigation model, jointly named as hypertext model in the WebML terminology.

Primarily, the hypertext model is targeted towards modelling, on the one hand, the network of navigations between the pages of certain web site, and, on the other hand, the composition of particular web pages from functional units, such as data unit, multi-date unit, index unit and many others, how is in detail described in WebML hypertext diagram notation documentation. An example of simple hypertext diagram is shown in (Fig. 1, example is taken from [11]).

Herein we put a very brief explanation of (Fig. 1); the example demonstrates a simple web site view denoted Product with four pages named: By category, Product details, Search and Images. All pages contain different units, for example, the page named Images contains the Product data unit and Enlarged images multi-data unit. But, we can also observe that there are the other units not included into any particular web page. These units are special process units, intended especially for data processing, but for other type of process activities as well.

We can point out, for example, login or logout units. As far as concerning the data process units, be specific, Add note, Add to cart and Update stock units, these units are especially target for inserting, modifying and deleting information records into database. Farther, we can observe that last two data process units are included in a special area named Add to cart transaction. This special type of area includes data process units subjected to database transaction.

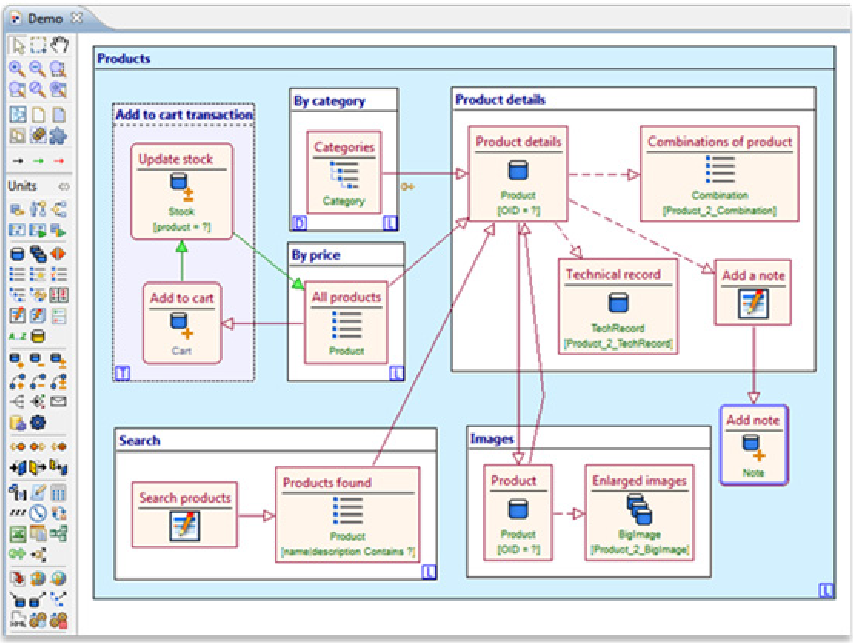


Fig. 1. WebML hypertext diagram - example

It is clear now that WebML diagram really can contain also process elements placed outside web pages on navigational paths between them. However, it has appeared that this combination of complex business logic and hypertext design of web site into one complex diagram is not a very efficient solution. Firstly, such diagram is exceedingly complex for easy understanding by developer; secondly, the process alias business portion of the system can be more easily modelled by standardized BPMN [3] notation with many advantages. In the next section we show how this idea of WebML and BPMN jointly modelling is implemented in WebRatio CASE tool.

4 WebRatio and BPMN

WebRatio [12] CASE tool is the one of the rarely existed CASE tools implementing the whole MDA process from beginning, i.e., from the data and process specification, to completion, i.e., to generation of the whole application. The last version 6.0 of WebRatio implements in one application both the WebML designer, and BPMN designer as well.

Herein we put a very brief explanation of the whole development process with the WebRatio CASE tool. The WebML method, as well as, WebRatio CASE tool insists on MDD (Model Driven Development) approach, it means that all life cycle of product development is controlled by models. The WebRatio Model Driven development environment allowing capturing business requirements in abstract model, i.e., Computationally Independent Model (CIM), in terms of Model Driven Architecture (MDA) paradigm as is defined by OMG (Object Management Group) and automatically generates a full-featured, industrial-strength, business application. The whole process is divided into three steps as follows:

- Design the Model – is a step of development process concerning with creation and verification of application model. This model is based on user requirements and constrains flowing from legacy situation, standards, company guidelines, etc.
- Customize the Rules – is a step of tailoring our model to better fulfil our requirements in more details. We can implement is this step, for example, a checking of input and output parameters, implement a constraining of different business criteria, add an online help, and so on.
- Generate the Application – in this step the WebRatio generates the whole application. It is the most complicated task, totally hidden from the view of end user. There is, of course, a possibility to affect the mode of generation by some parameters.

Now we can focus on the first step of development, i.e., on the step of designing the model. We can observe that WebRatio includes two quite different diagrammatic instruments, (1) the first of them is a designer of standardized BPMN diagrams targeted modelling of business processes defined by the precedent analysis of business requirements, and (2) second, the special designer of WebML web-method modelling notation, targeted modelling navigations between web site pages, content of these pages by means of many specialized WebML units, and eventually processes others than processes follow from business requirements. The first type of model, i.e., the process model, strongly based on BPMN notation, is used to define:

- Organisation and roles
- Activities and assignments
- Business roles
- Business workflow

The second model, i.e., the application model, is based on WebML modelling language, and is used to define:

- Page contents
- Business logic

- User interface & Visual identity
- Integration

In the frame of whole development process the construction of the first model precedes the second one. By the Author's opinion, this conception of utilisation of two very different types of model, managed by two different and independents subjects is very inventive. This concept also shows quite clearly a difference between a business and informational process, the subject, not very frequently discussed generally.

There are four different perspectives of WebRatio tool development process as illustrated in the following text. We can observe that there are four different roles participating on development process, namely:

- Business analyst – creating a business model in BPMN based on users requirements
- Application/WebRatio Analyst – creating an application model in WebML based on users requirements, web site design guidelines and just created foregoing business model as well
- Web Designer – elaborating and completing the application model, as well as creating layout templates for all visual units of the system in order to produce a clear and well-arranged graphical appearance of all web pages of the web site
- Java Developer – developing and implementing custom components needed to implement all non-standard content, hybrid and process units of the model

The concept of Model-Driven Development Process implemented in WebRatio CASE tool consists from following four steps. We start with a specification of traditional BPMN process model. In the second step we manually enrich and extend this model to create a choreography model (extended BPMN). Further, we run an automatic transformation of BPMN to WebML, then; we possibly repeatedly manually refine produced WebML models. And finally, we run automatic code generation on J2EE platform. The whole process is shown in the following.

1. Business process (BPMN) specification
2. Choreography model (extended BPMN)
3. Application Model (WebML)
4. Generating Application (J2EE)

In the following we try explaining the most important step of the whole application development; a transformation from BPMN scheme to WebML scheme. The main question we need to explain herein is as follows, how we can generate the whole application specified in WebML from BPMN diagrams? To start we suppose following simple BPMN diagram (Fig. 2, example is taken from [11]) presenting a simple BPMN diagram of Product Catalogue Application. We see a business process borne by three business participant: Employee, Supervisor and Treasurer. The Employee performs Edit Report manual task, The Supervisor performs Review

Report manual task and Treasurer performs Confirm Receipts manual task and Process Expanses service task.
The main principles of transformation process from BPMN to WebML are as follows:

- The manual tasks and gateways are not transformed at all
- The service tasks and gateways are transformed to operation module in WebML
- The user tasks and gateways are transformed to hybrid module in WebML
- There is generated one control site view for BPMN pool
- There is generated one site view for one BPMN line

The result of such transformation of Confirm Receipts manual task is shown in (Fig. 3). We see that the Confirm Receipts task is transformed into Confirm Receipts web page in WebML model with three WebML units: Info, Get Parameters and User Input. We can also observe the process units needed for input and output parameters passing, located in the diagram outside the Confirm Receipts web page.

In conclusion, the whole transformation of BPMN model into WebML model implemented in WebRatio CASE tool is complex and sophisticated, but really implementing the Model Driven Development paradigm in its entirety.

5 Our Approach

The approach we used to create a design support for inclusion of social features into a modelling and developing phase of Enterprise Systems, lies in the possibility to build custom units in WebRatio CASE tool [13]. By means of these custom units it is possible to extend a WebML modelling notation to fulfil our needs. Our extensions comprise the units dealing with some social aspects of business process management as discussed before.

WebML is built on very few, highly configurable concepts, which can be used to effectively analyse and design complex web applications for publishing, updating or processing its content. The key aspect of WebML is a capability to define a hypertext model of a web site consisting of pages, content units, and operation units, linked each other to form a design specification of the whole system.

However, the core units provided by WebML may not be sufficient for covering the entire spectrum of application requirements; moreover one may need to use his own specialized software components in the course of a web applications development. To support this purpose, WebRatio enables developers to implement so called *custom units* (also called *plug-in units*). A custom unit can be of type of *content unit* or *operation unit* and is defined by a developer, and is not, of course, included in the WebRatio standard installation. Thus, a custom unit is both similar and different to the standard unit.

A development of a custom unit covers all aspects of WebRatio CASE tool architecture, because a definition as well as an execution of the unit requires addressing both design-time and run-time issues.

WebRatio CASE tool let developers to use their custom units in the course of web site development and these units will cooperate perfectly with standard units too. Since predefined WebML units are implemented in the same way as user's custom units, a user can start familiarising with the components of a custom unit by looking at those of the standard units.

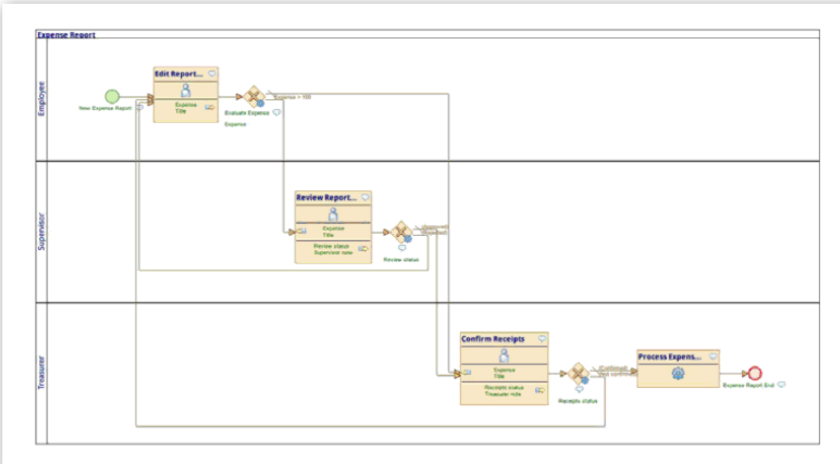


Fig. 2. BPMN model of the Product Catalogue Application

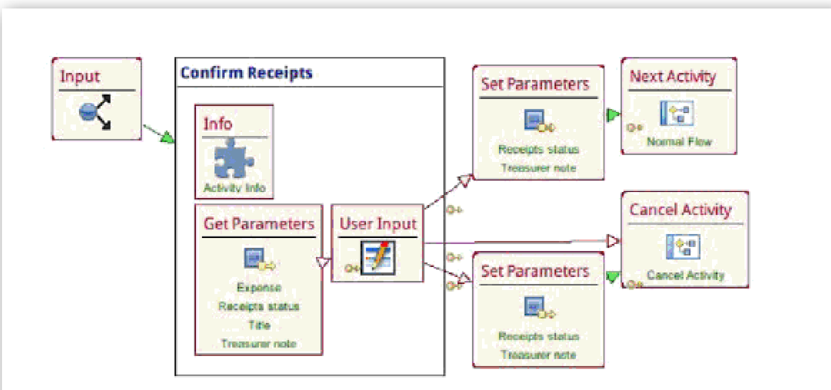


Fig. 3. Generated WebML hypertext for the Confirm Receipt activity

6 Social Features Implemented in WebRatio

At the beginning, we have introduced a question of what kind of social BPM features our work should support (development phase or run-time?). The final decision at the moment is to implement only run-time features. It gives the added value to WebRatio applications developers so their web systems will support BPM social features just by adding new units with abilities as follows.

- Login to the social network
- Add a comment/note to the process as anonymous.
- Add a comment/note to the process as logged user.
- Delete a comment/note from the process for logged user.
- Read a list of comments for the current process.
- Read a list of all comments for the current application.

We also have captured non-functional requirements are stated below.

- We are aimed on implementing social BPM features from run-time phase of an application life cycle.
- We must provide functionality for Facebook API.
- We need an extensibility of new social networks.

One of the ways how to extend functionality of WebRatio CASE tool is an utilisation of the concept of Custom Units (see preceding section). The concept of custom units is easily accessible to every user of WebRatio CASE tool. At the present we started with implementation of custom units covering only Facebook social network. Slight modification is needed to support other social networks (Twitter, LinkedIn). This modification only depends on API provided by particular social network provider. We have decided to implement only following functional types of custom units:

Get Unit

- Read a list of comments for the current process.
- Read a list of all comments for the current application.

Post Unit

- Add a comment/note to the process as anonymous.
- Add a comment/note to the process as logged user.

Remove Unit

- Delete a comment/note from the process for logged user.

An example of some Social Units included into a Master Page of synchronised BPM project is shown in Fig. **Fig.** 4. Subsequently, the Fig. 5 shows the login page for BPM testing scenario created for the purposes of project testing. Finally, the Fig. 6 shows how particular social units are included in the process detail of *Open and Check CV* activity of our testing project.

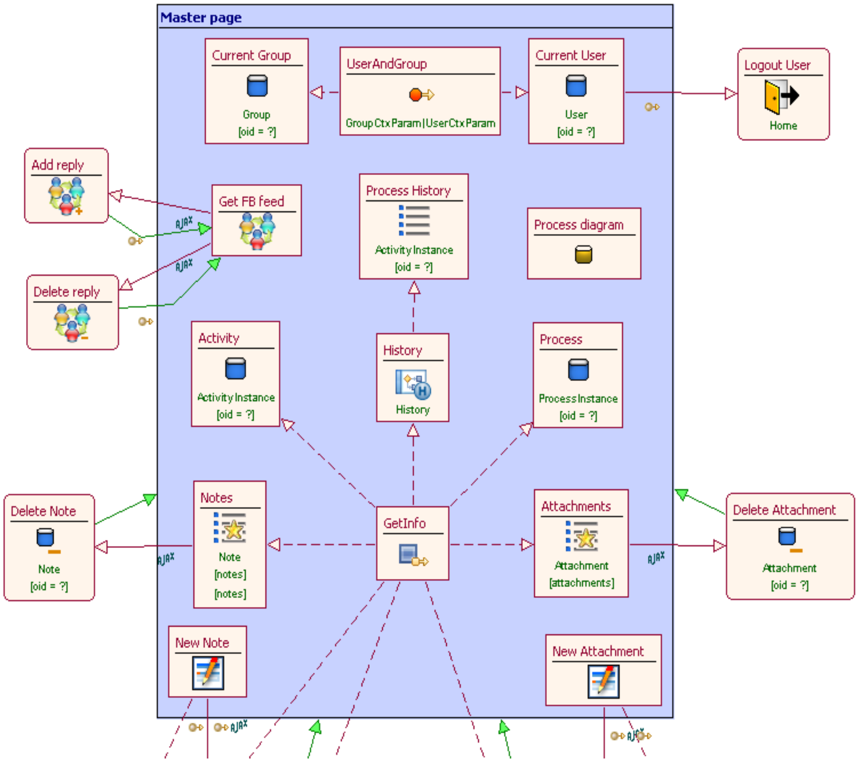


Fig. 4. Units implemented in BPM testing scenario - master page

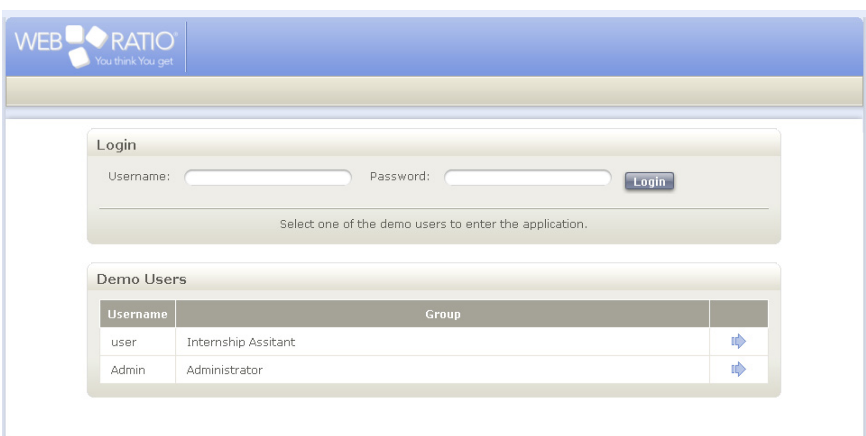


Fig. 5. Units implemented in testing scenario - login page

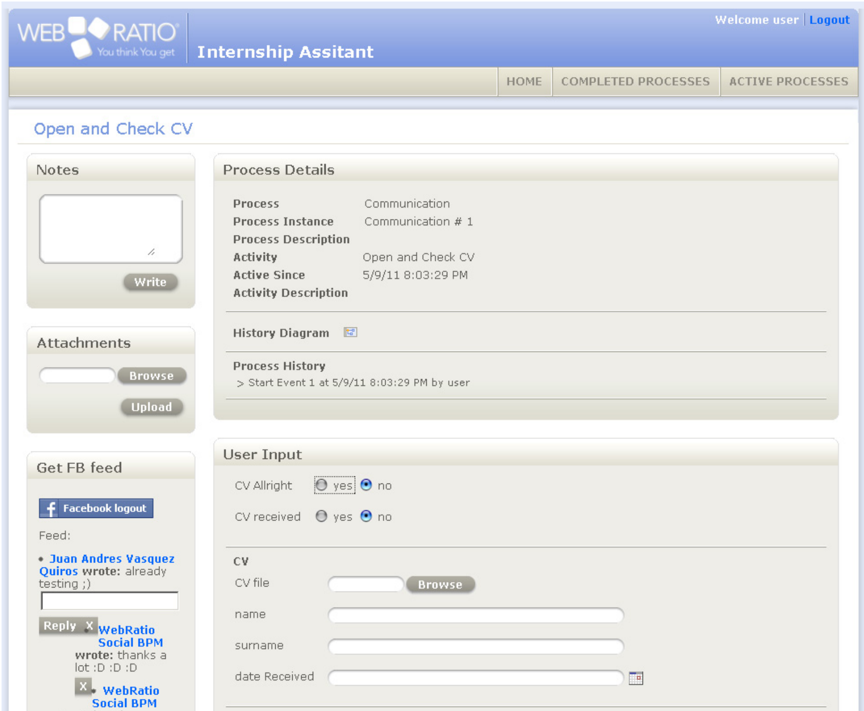


Fig. 6. Units implemented in testing scenario - process details with Social Units

7 Conclusion and Future Work

This work is partially realised on an output of the master thesis of Dominik Franěk [14], supervised by the Author of this contribution. Moreover, many ideas have been initiated as a result of discussion with a WebRatio development team member, Mr Marco Brambilla.

Planned future extensions cover a support for other social networks such as Twitter, LinkedIn. At present, only Facebook social network is supported and there is a strong need to add all popular social networks too.

Another direction of our future work would be an automatic generation of Social Units during a synchronisation of BPM project with Web project. But, this part can't be done without modifying a WebRatio source code and only in cooperation with WebRatio developer's team has it would be realised.

To summarize presenting article; we firstly explain the present-day necessity to enrich Business Process Management of Enterprise System of social-network features. Afterwards we show connections between WebML method and WebRatio CASE tool on one side and BPMN notation and social features on other side.

Further, we briefly describe a process of transformation from BPMN scheme to WebML scheme. Furthermore, we describe our implementation of some custom social units created for WebRatio CASE tool. Finally, we propose some directions of future Author's works.

Acknowledgement. This research (work) has been partially supported by Ministry of Education, Youth and Sports of Czech Republic under research programs number MSM6840770017 and SGS projects number SGS10/267/OHK3/3T/13.

References

1. WebML at Wikipedia, `http://en.wikipedia.org/wiki/WebML`
2. WebML method, Home Page, `http://www.webml.org/webml/page1.do`
3. BPMN specification at OMG site, `http://www.bpmn.org/`
4. Brožek, J., Merunka, V., Merunková, I.: Organization Modeling and Simulation Using BORM Approach. In: Barjis, J. (ed.) EOMAS 2010. LNBIP, vol. 63, pp. 27–40. Springer, Heidelberg (2010)
5. Pergl, R.: Supporting Enterprise IS Modelling Using Ontological Analysis. In: Barjis, J., Eldabi, T., Gupta, A. (eds.) EOMAS 2011. LNBIP, vol. 88, pp. 130–144. Springer, Heidelberg (2011)
6. Becker, J., Rosemann, M., von Uthmann, C.: Guidelines of Business Process Modeling. In: van der Aalst, W.M.P., Desel, J., Oberweis, A. (eds.) BPM 2000. LNCS, vol. 1806, pp. 30–49. Springer, Heidelberg (2000)
7. Ko, R.K.L., Lee, S.S.G., Lee, E.W.: Business process management (BPM) standards: a survey. Business Process Management Journal 15(5), 744–791 (2009)
8. OMG. Business Process Model and Notation (BPMN) Specification 2.0. Object Management Group Inc. (March 2011)
9. InfoQ: Is There Social BPM?, `http://www.infoq.com/news/2010/05/SocialBPM` (citated May 15, 2011)
10. Social BPM: Is It Social, or is It BPM? - ebizQ Forum, `http://www.ebizq.net/blogs/ebizq_forum/2010/03/social-bpm-is-it-social-or-is-it-bpm.php` (citated May 15, 2011)
11. Butti, S.: A MDD approach to BPM, presentation online, `http://www.slideshare.net/stefano_butti/webratio-a-mdd-approach-to-bpm-4872510`
12. WebRatio, Home page, `http://www.webratio.com`
13. Acerbis, R., Bongio, A., Brambilla, M., Butti, S.: WebRatio 5: An Eclipse-Based CASE Tool for Engineering Web Applications. In: Baresi, L., Fraternali, P., Houben, G.-J. (eds.) ICWE 2007. LNCS, vol. 4607, pp. 501–505. Springer, Heidelberg (2007)
14. Franěk, D.: Social BPM Features in Webratio tool, Master Theses, Czech Technical University in Prague (2011)

The Function-Behaviour-Structure Diagram for Modelling Workflow of Information Systems

Stanislaw Jerzy Niepostyn and Ilona Bluemke

Institute of Computer Science, Warsaw University of Technology,
Nowowiejska 15/19, 00-665 Warsaw, Poland
{S.Niepostyn,I.Bluemke}@ii.pw.edu.pl

Abstract. Currently, no single UML diagram provides the satisfactory completeness and consistency of the system description. There is also no BPMN diagram to satisfy such requirements. The satisfactory completeness means that the model enables to describe fully a function, a structure, and a behaviour of the IT system. With BPMN diagram one cannot provide a complete data model i.e. the structure of the IT system. The proposed Function-Behaviour-Structure activity diagram introduced in this paper enables to develop consistent and satisfactorily complete models.

Keywords: MDA, Business Process Management, UML, BPMN, activity diagram, consistency, completeness.

1 Introduction

In the beginning of the year 2001 the Object Management Group (OMG) launched Model Driven Architecture [1] based on transformational approaches (generating models from other models). The main concept of the MDA standard specifies rules for building application which are generated from models at higher (business) level of abstraction. Despite a lot of research conducted during many years MDA tools still have not been produced on an industrial scale. The cause of such situation is mainly resulting from lacking rules defining how the elements from different models relate to each other. The reason for the lack of such rules is the semantic of the UML [2] models defined in natural language. UML cannot provide a straightforward way of representing a connector (association) and there is no specific construct for representing architectural styles [3]. UML cannot fully define the relationships between diagrams so the consistency across diagrams must be ensured manually [4] or expressed e.g. in OCL [5].

Model Driven Architecture is in the stagnation for the last few years while Business Process Management (BPM) [6] has been quickly developing. BPM systems support execution of business processes (workflow applications) according to BPM concepts [7]. BPM platforms also allow users to build and deploy business process models supported with Business Process Execution Language for Web Services (BPEL) [8] and XML Process Definition Language (XPDL) [9] standards. Today's

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 425–439, 2012.

BPM platforms offer their own notations to describe business processes. Therefore, since January 2011, OMG proposed the Business Process Model & Notation (BPMN) [10] language to build and deploy business process models on BPM platforms. It can be noticed that BPMN ensures the consistency of the business processes (workflow). BPMN provides only the description of a system behaviour and the description of its structure and function are neglected. The lack of sufficient structure description of the system can become soon a significant restriction to the full automation in BPM solutions.

In this paper we introduce a new Function-Behaviour-Structure Activity diagram, which enables to keep the consistency and satisfactory completeness of the application model. Some model elements are formally described in Z notation [11, 11] in section 3 to provide the consistency proof. In section 2 the dimensions of software architecture are described, then completeness and types of inconsistencies are described. The rationale of applying the FBS activity diagram in the design view of the software architecture to derive the complete and consistent UML model is given in section 4. Related work and some conclusions are given in sections 5 and 6.

2 Incompleteness and Consistency of Model

According to Function-Behaviour-Structure (FBS) framework introduced by Gero [13] the purpose of the design's description is to transfer sufficient information about target system so it can be constructed. The description must at least enable to incorporate a function, a structure, and a behaviour of the target system. Therefore the development of software in which one cannot take into account these three dimensions, are "doomed to fail". Truyen [14] described a model, in major MDA concepts, as a formal specification of the function, structure and behaviour of a system. He claims, that model must be represented by a combination of UML diagrams. Spanoudakis and Zisman [15] described this as a situation, in which model inconsistencies may arise.

Below we explain informally the model consistency, which we subsequently apply in the analysis of selected diagrams. Then we present our concept of the dimensions of the software architecture which form consistent description of software architecture.

2.1 Model Inconsistencies

To assert that something is consistent we have to declare what it is consistent with. Software models describe system from different points of view, at different levels of abstraction and granularity, in different notations. They may represent viewpoints and goals of different stakeholders. Usually inconsistencies between diagrams are arising. Inconsistencies reveal design problems. The roots of consistency can be found in formal methods. The research on consistency models was started by Finkelstein [16]. Finkelstein stated, that inconsistency is not necessarily a bad thing, and should be evaluated after the translation of the model specification into formal logic. UML is not a formal language so often UML models are translated into more formal notation. UML is widely used in the software design so the problem of inconsistency in UML models received special attention. In UML inconsistencies between class, state machine and sequence diagrams [17] are studied. Inconsistencies arise because some models are overlapping [15]

UML consistency analysis goes far beyond checking syntax and semantics, it should also encompass other domains like targeted programming language, modelling methodology, modelled systems, and application and implementation domains.

Mens [18] proposed five consistency types:

1. Inter model (vertical) consistency. Consistency is evaluated between different diagrams and different levels of abstraction. The syntactic and semantic consistencies are also taken into account.
2. Intra model (horizontal) consistency. Consistency is validated between different diagrams but at the same level of abstraction.
3. Evolution consistency. Consistency is validated between different versions of the same UML diagram.
4. Semantic consistency. Consistency is validated for the semantic meaning of UML diagram defined by an UML metamodel.
5. Syntactic consistency. Consistency is validated for the specification of UML diagrams in an UML metamodel.

Another classification divides consistency into static and dynamic. Static consistency can be verified without running the model while dynamic constraint cannot be verified until runtime. In [19] a survey of consistency checking techniques for UML models is presented. The existing techniques are classified based on intermediate representation into three categories: formally represented, extended UML - intermediate representation is defined as an extension in UML diagrams and without intermediate representation. Many interesting information on consistency problems in UML-based software development can be found in [20].

2.2 Dimensions of the Software Architecture

In the majority of projects using UML diagrams [21, 22], use case diagrams are developed at the beginning of software development to describe the main functions of the software-based system. Then class diagrams are created to show the structure of the system, and state machine diagrams are built to show the behaviour of system's elements ([23, 24]). Subsequently activity or sequence diagram can be used in order to verify consistency of other diagrams. These diagrams are also using visualizing scenarios i.e. – use case realization diagrams.

Activity diagram enables to associate activities with objects (instantiate classes), and use-cases ([23, 24, 25]). It can be noticed that Use Case, Class and State Machine diagrams are orthogonal (Fig. 1), and enable to derive use case realization diagram [26]. A model, which adequately integrates these diagrams thus enables to keep the consistency and the satisfactory completeness of the whole system because these three diagrams do not have common elements. Anyone can interpret the operation of the class (dimension of the structure), the state in statechart (dimension of the behaviour), and use case (dimension of the functionality) as a single element of the integrated model. Such integrated model (diagram) enables to achieve *satisfactory* completeness. We define *satisfactory* completeness as comprising necessary elements (listed above) and at least one element that integrates all those three dimensions of the software architecture.

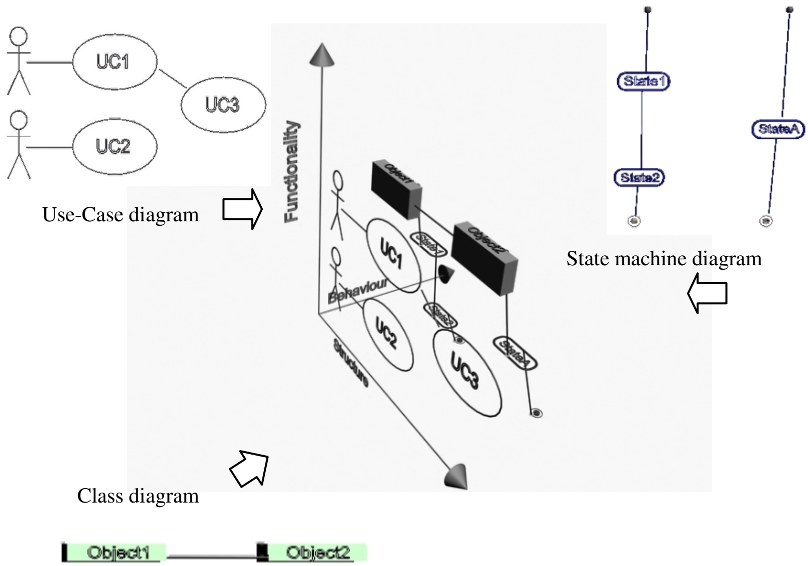


Fig. 1. Three dimensions of the software architecture view

3 Analysis of Some Diagrams

In this section we analyse the consistency and the satisfactory completeness of class, state machine and use case diagrams. In this analysis we use simplified metamodels, without cardinalities, and Z schemas describing some metamodel's elements used in the consistency reasoning. Cardinalities do not change the results of consistency analysis. Next, we show that other diagrams like activity, sequence or BPMN diagrams do not have properties required to sufficiently describe the target system so we propose such a diagram in section 4.

3.1 Class Diagram, State Machine Diagram, and Use Case Diagram

The simplified metamodel of the class diagram is presented in Fig. 2.a and its formalization using Z schemas is shown in Fig. 2.b.

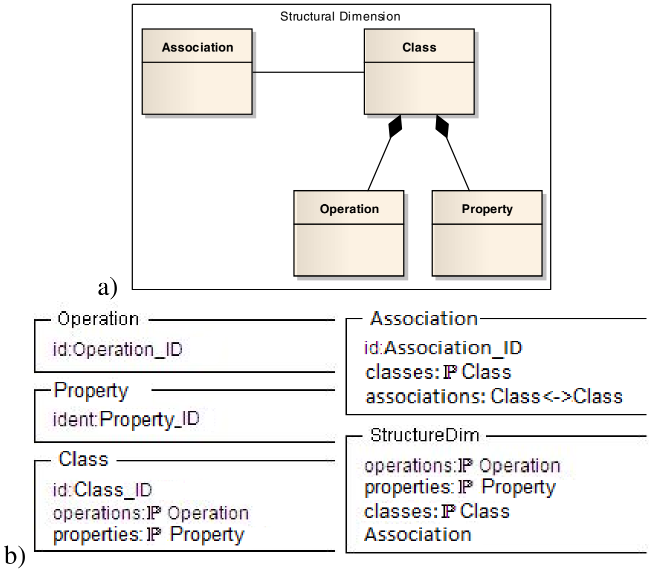


Fig. 2. a) UML metamodel of a class b) Z formalisation of the class diagram

In the schema *StructureDim*, classes are the set of *Class* instances, operations are the set of *Operation* instances, and properties are the set of *Property* instances. Class diagram describes only the dimension of the structure of the system. All elements describe the structure of the system but some behaviour properties could be generated from *Operation* [27].

The simplified metamodel of the state machine diagram is presented in Fig. 3.a and its formalization is shown in Fig. 3.b. In the schema *BehaviourDim* states are the set of *State* instances. State machine diagram describes only the dimension of the behaviour of the system.

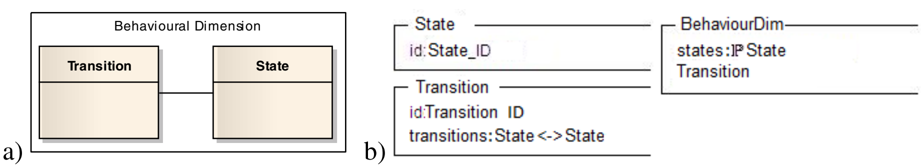


Fig. 3. a) UML metamodel – state machine diagram, b) Z - formalisation of the statechart - dimension of behaviour

The use case diagram (dimension of functionality) is presented in Fig. 4.a, and its formalisation in Z schemas is shown in Fig. 4.b. In the schema *FunctionalDim* use cases are the set of *UseCase* instances and actors are the set of *Actor* instances. Use case diagram describes only the dimension of the functionality of the system. *Actor* and *UseCase* are defined in the standard UML as the elements that describe the behaviour rather than the functionality of the system.

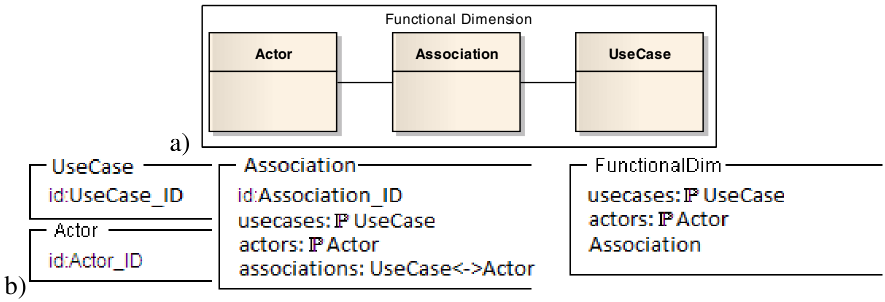


Fig. 4. a) Metamodel of the use case, b) Z - formalisation of the dimension of functionality

Comparing all diagrams (dimensions) with each other no common element can be found so those three UML diagrams must be consistent. Moreover, those three UML diagrams could describe satisfactory completeness of an IT system if the operation, the state, and the use case elements are integrated into a single UML diagram.

3.2 Activity Diagram

The activity diagram is presented in Fig. 5.a and its formalisation in Z schemas is shown in Fig. 5.b.

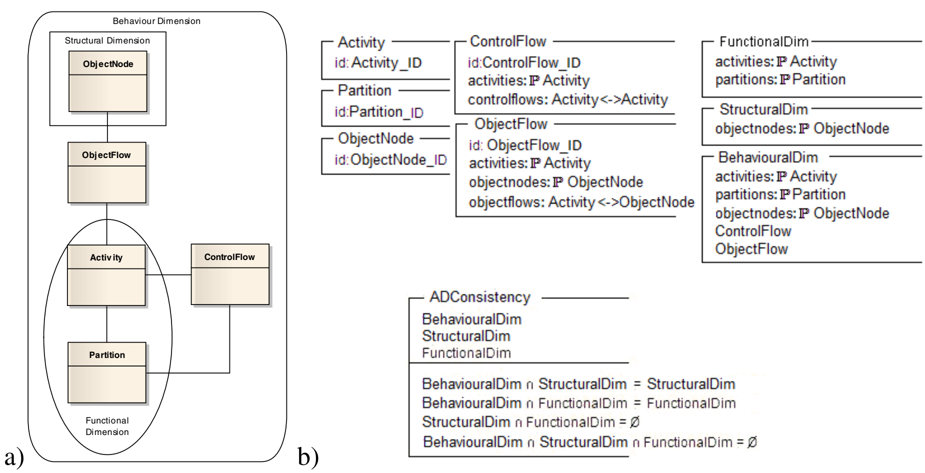


Fig. 5. a) Metamodel of the activity diagram, b) Z - formalisation of the activity diagram

The dimension of the functionality describes *Partition* and *Activity*. *ObjectNode* represents the dimension of the structure, and the dimension of the behaviour contains all elements of the activity diagram. There is no common element in all three dimensions. Thus it is not possible to integrate the three dimensions of software architecture in this activity diagram. The dimension of the behaviour has common elements with the dimension of the structure, and also with the dimension of the functionality. It means, that the other elements of the activity metamodel are dependent on each other so the three dimensions overlap with each other. According to Spanoudakis and Zisman [15] “inconsistencies arise because the models overlap” therefore, the three dimensions of the activity model are not consistent [28]. This property implies also that the corresponding UML models (use case diagram, state machine diagram, class diagram) may not be consistent.

3.3 Sequence Diagram

The sequence diagram is presented in Fig. 6.a, and its formalisation in Z schemas is shown in Fig. 6.b. The dimension of functionality describes *lifelin*e. Lifeline and *message* represent the dimension of structure, and the dimension of behaviour contains all elements of the sequence diagram. There is a common element in the three dimensions so it is possible to integrate the three dimensions of software architecture in this sequence diagram. The other elements of the interaction metamodel are dependent on each other so the three dimensions overlap with each other. According to Spanoudakis and Zisman [15] inconsistency definition the three dimensions of the interaction model are not consistent. This property implies also that the corresponding UML models (use case diagram, state machine diagram, class diagram) may not be consistent.

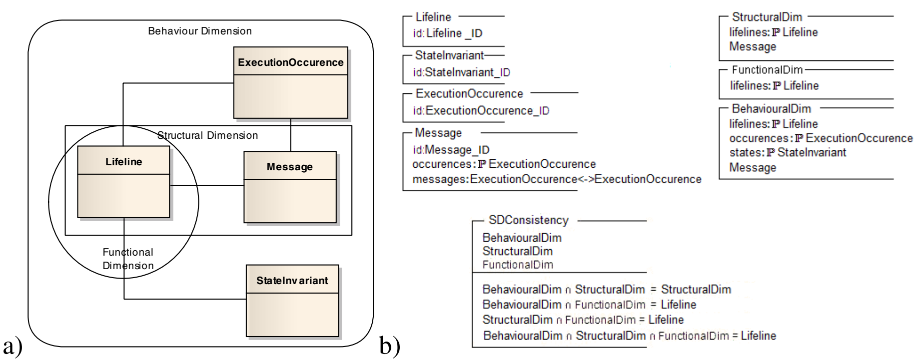


Fig. 6. a) Metamodel of the sequence diagram, b) Z - formalisation of the sequence diagram

3.4 BPMN Diagram

The BPMN metamodel is presented in Fig. 7.a and its formalisation in Z schemas is shown in Fig. 7.b.

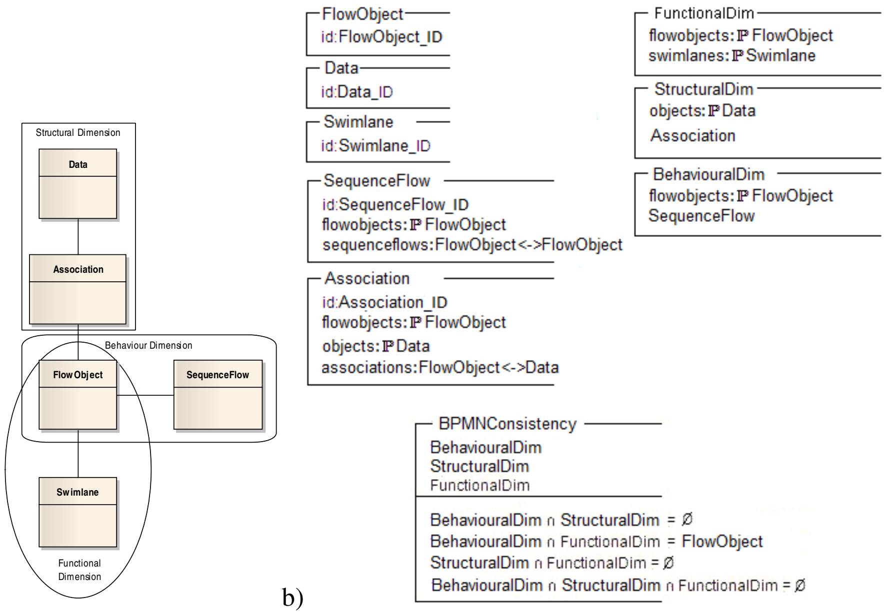


Fig. 7. a) Metamodel of the BPMN diagram, b) Z - formalisation of the BPMN diagram

The dimension of functionality describes *Swimlane* and *FlowObjec*t. *Data* and Association represent the dimension of structure, and the dimension of behaviour contains *FlowObject* and *SequenceFlow* elements of the BPMN diagram. As no common element is present in the three dimensions, it is not possible to integrate the three dimensions of software architecture in this BPMN diagram. It means that BPMN model does not have properties to satisfactory describe the target system. It can be noticed that the dimension of the structure is unsatisfactory to perform mapping of a class diagram to *Data* and *Association* elements. Therefore with BPMN diagram one cannot provide a complete data model.

4 FBS Activity Diagram

The FBS activity diagram enables to build a model integrating the three dimensions of software: functional, structural and behavioural. In Fig. 8 an example of a routine task in an office modelled by FBS activity diagram is shown.

The header of the diagram describes the objects and the first column shows the Actors. In following columns the activities are given, each one is performed by an appropriate actor. There are several kinds of the activities: `Creating`, `Checking`, `Archiving`, `Approving` and `Other`. These activities have the incoming and outcoming instances of the classes. Figure 8 presents a request of a service from an office. A *Customer* fills a written request (Creating request), then *Clerk* checks this request (Checking request). After this checking, the *Clerk* looks into it (Creating opinion). The *Supervisor* accepts the request (Approving opinion and request) and

Clerk archives his decision (Archiving request and opinion). Then the *Clerk* prepares the reply (Creating reply), the *Supervisor* accepts it (Approving reply) and, at the end, the *Customer* receives it (Receiving reply).

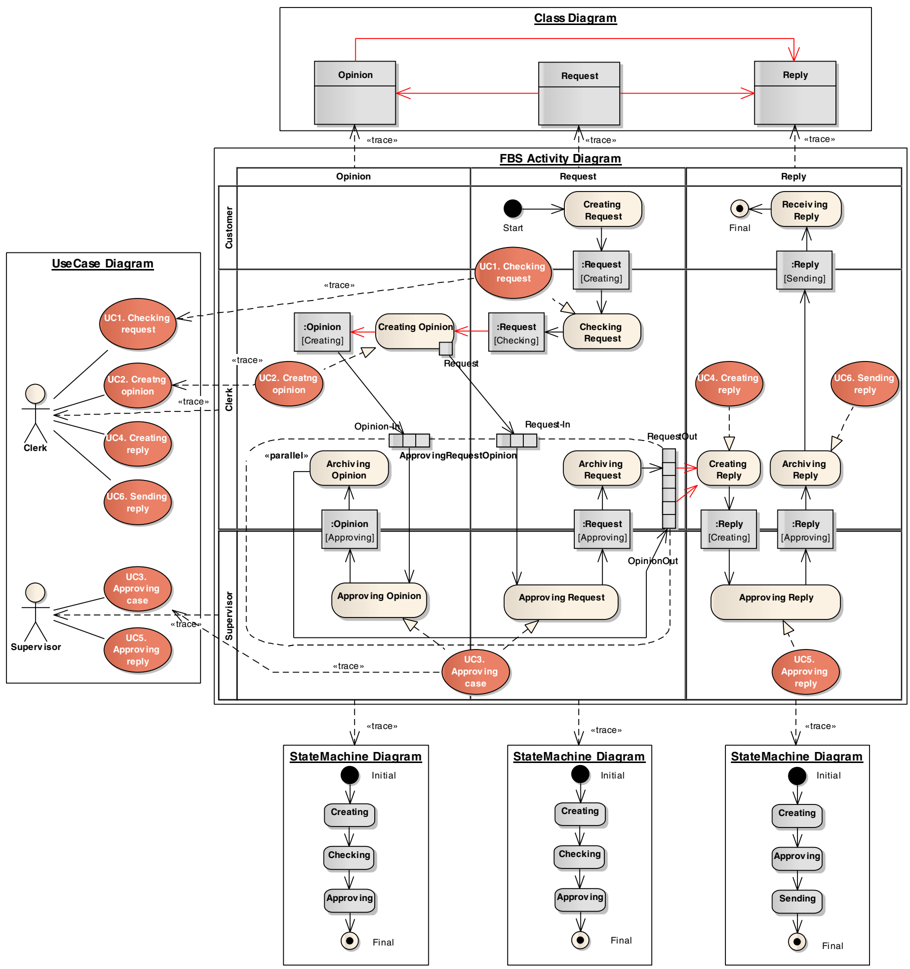


Fig. 8. The Function-Behaviour-Structure activity diagram

In Fig. 8 the mappings between FBS activity model and UML diagrams are also shown. These diagrams describe the design view of the software architecture. The FBS activity model is in simple and unambiguous relationships with class diagram (structure), state diagram (behaviour), and use case diagram (functionality).

Each element in the header of the FBS activity model corresponds to only one object, which is instance of a proper class from the class diagram. The associations between objects are derived from the edges of horizontal object flow (in red colour). Moreover, each FBS object has simple and unambiguous state diagram. Each FBS object with its state corresponds to only one state in the state diagram. Transitions in this state chart are derived from FBS activity with horizontal object flow between FBS objects. In similar way the FBS activities can be mapped onto use case diagram.

A few FBS activities are realized by one use case, and each use case is associated with an actor in use case diagram. The *Actor* is derived from the horizontal *Partition*, which is grouping particular FBS activities. In order to improve the readability of Fig. 8, not all dependencies between diagrams are visible.

4.1 Satisfactory Completeness of the FBS Activity Model

In Fig. 9 the simplified UML activity FBS meta-model is presented. The dimension of functionality describes *Actors*, *Use Cases*, and *Activities*. *States of Objects* with *Activities* and *verticalObjectFlow* represent the dimension of behaviour, and the dimension of structure contains *Objects*, *horizontalObjectFlow* and *Activities*.

Z-formalization of the FBS activity diagram is shown in Fig. 10. The common element of the three dimensions is *Activities,* it could be used to integrate the three dimensions of software architecture in this diagram. Other elements of the FBS model fully describe the three dimensions so the FBS activity diagram is satisfactorily complete.

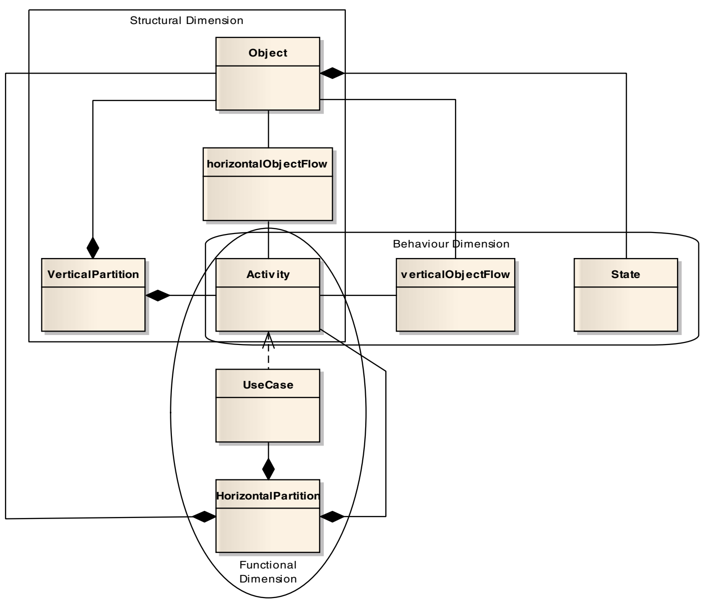


Fig. 9. Metamodel of the FBS activity diagram

```
Activity
id: Activity_ID

hPartition
id:hPartition_ID

vPartition
id: vPartition_ID

Object
id: Object_ID

UseCase
id:UseCase_ID

State
id: State_ID

vObjectFlow
id:vObjectFlow_ID
activities: ℙ Activity
objectnodes: ℙ Object
vobjectflows: Activity <->Object

hObjectFlow
id:hObjectFlow_ID
activities: ℙ Activity
objectnodes: ℙ Object
hobjectflows: Activity <->Object

FunctionalDim
activities: ℙ Activity
hpartitions: ℙ hPartition
usecases: ℙ UseCase

StructuralDim
objects: ℙ Object
activities: ℙ Activity
vpartitions: ℙ vPartition
hObjectFlow

BehaviouralDim
activities: ℙ Activity
states : ℙ State
vObjectFlow
```

Fig. 10. Z-formalisation of the FBS activity diagram

```
FBSConsistency
BehaviorDim
StructureDim
FunctionalDim

BehaviorDim ∩ StructureDim ∩ FunctionalDim = Activity
HorizontalPartition ∩ BehaviorDim = ∅
HorizontalPartition ∩ StructureDim = ∅
UseCase ∩ BehaviorDim = ∅
UseCase ∩ StructureDim = ∅
State ∩ FunctionalDim = ∅
State ∩ StructureDim = ∅
VerticalPartition ∩ FunctionalDim = ∅
VerticalPartition ∩ BehaviorDim = ∅
Object ∩ FunctionalDim = ∅
Object ∩ BehaviorDim = ∅
```

Fig. 11. Formalization of consistency of the FBS activity model

4.2 Consistency of the FBS Activity Model

Inconsistencies arise between elements belonging to several models. In Fig. 11. Z-formalization of the consistency of FBS activity diagram is presented. The common element of the three dimensions is *Activity* and others elements are not dependent on each other so the three dimensions do not overlap with each other. Therefore, according to Spanoudakis and Zisman [15], the three dimensions of the FBS activity model are consistent. This property implies that the corresponding UML models (use case diagram, state machine diagram, class diagram) are consistent too. Any change in the *Activity* element is visible in all dimensions of the FBS activity model. The changes of other elements of the FBS activity model do not influence each other. Z-schema *FBSConsistency* (Fig.11) describes the above mentioned rules.

5 Related Work

Different software models describe the same system from different points of view, at different levels of abstraction and granularity, possibly in different notations. They may represent the perspectives and goals of different stakeholders. Usually some inconsistencies between models are arising. Inconsistencies in models reveal design problems. If these problems are detected at the early stages of the design, costs of fixing them are much lower than if they are detected at later stages of software design.

Usually UML models are translated into programming languages. Inconsistent UML model may result in an imprecise code. Inconsistencies highlight conflicts between the views and goals of the stakeholders, indicating those aspects of the system which should be analysed.

The approach to model or describe the system in three dimensions i.e. function, structure and behaviour is widely used. E.g. Goel, Rugaber, and Vattam proposed in [29] the structure, behaviour, and function modeling language. They viewed SBF as a programming language with specified abstract syntax and static semantics. The SBF language captures the expressive power of the programs and provides a basis for interactive construction of SBF models. They also described an interactive model construction tool called SBFAuthor that is based on the abstract syntax and static semantics of the SBF language. The precise specification potentially enables a range of additional automated capabilities such as model checking, model simulation, and interactive guides and critics for model construction. The problems of consistency and completeness of model are not discussed in their paper.

The Integrated Notation for specifying software architecture introduced in [30] also proposes three levels of abstraction i.e.: structure (specified by graphical box and line diagram), behavioural specification using Input/Output Automata and abstract data types (ADT) described by Larch traits. Bastarrica, Ochoa and Rossel claim that starting from the structure, refining it with behavioural details and using abstract data types the software architect can obtain a consistent model. The Integrated Notation does not have elements which are common for several levels so the rules to keep consistency among layers [30] are much more complicated than in the FBS activity model. The Integrated Notation does not meet our preceding definition of satisfactory completeness.

Vondrak presented in [31] the Business Process Studio application based on functional, behavioural, and structural views. He applied the object diagram to structural specification, state diagram to describe behaviour of the system, and dataflow diagram as the functional dimension. In this approach a c*oordination model,* based on functional and object models, is used to show *how* the process will be enacted. The coordination model specifies interactions among objects (active and/or passive) and defines the way all these activities are synchronized based on principles used in Petri Net. The coordination view is the most important because it enables to define the execution order of all activities, including conditions for their potential concurrency. It means that the correct order is defined, as well as sharing of used resources. Each activity can have more than one scenario with the duration time and costs associated to provide necessary information for the analysis. Based on the architecture definition captured in a functional model, the "primitive" activities are accompanied by sub-processes icons that can be refined further into more detailed collaboration models again.

Method that would allow to derive the software architecture of any system based on its analysis model was proposed by Elleuch, Khalfallah, and Ahmed in [32]. For that purpose, they introduce a new layer to the Model Driven Architecture (MDA) that takes into account the software architecture. The analysis model is termed the Architecture Independent Model (AIM), which is compliant to the UML 2.0 metamodel. They consider the software architecture in the Architecture Specific Model (ASM), which complies to the defined architectural meta-model. The mapping of AIM into ASM is conducted by using the both meta-models. In this approach only the dimension of the structure based on the class diagram is used. Authors did not explore the incompleteness or inconsistency in their model ArchMDE.

UML is the notation for software engineering projects and many adequate software systems are built with its use. The incompleteness and the inconsistency allowed by UML are a source for problems in the software development process. An interesting question is to what degree inconsistency and incompleteness in UML designs impact software engineering projects. To answer these questions Lange et al. developed a number of techniques for analyzing UML designs. In [33] they attempt to quantify inconsistency and incompleteness of UML diagrams. In this article the analysis is focused on the four most widely used types of diagrams: class diagrams, state chart diagrams, use case diagrams and message sequence charts. Authors did not take into account the dimensions of the software architecture, but they formed some hypothesis about incompleteness and inconsistency in UML diagrams. They performed a number of experiments based on industrial case studies. From these experiments they observe that quantifying inconsistencies and incompleteness provides insight into the use of UML. Although no reference numbers have been established yet, the absolute number of inconsistencies in UML designs is quite large. They also noticed that the types of inconsistencies appear strongly related to the habits and conventions used by the designers.

6 Conclusions

In this paper we have presented a new Function-Behaviour-Structure activity diagram which has several advantages. Our diagram enables to keep the consistency and satisfactory completeness of the application's model. The FBS activity diagram allows to automatically generate complete workflow applications with no need for any "manual" programming. In addition, we have shown that the UML diagrams mapped from the FBS activity model are consistent.

The practical usage of FSB diagram may be questioned. The presented in Fig. 8 FSB diagram, prepared for six use cases, was not "easy to understand and read". In industrial projects the number of use cases is significantly greater but usually complex models are decomposed into submodels. Such approach is commonly used for UML models and also can be applied for FSB diagram. FSB activity diagrams were successfully applied in several industrial realization of IT systems in Poland.

In the design process UML models are refined and to keep the consistency among them, many complicated techniques are used e.g. [20,34]. Instead, it might be considered, to refine the FBS activity model and consecutively map it to the consistent UML diagrams.

The next step in our work is to develop the tool automatically generating complete workflow applications based on FSB activity diagram.

Acknowledgments. We are very grateful to the reviewers for many valuable remarks.

References

1. OMG Model Driven Architecture, `http://www.omg.org/mda`
2. Unified Modeling Language: Superstructure, version 2.4.1, formal (August 05, 2011), `http://www.omg.org/spec/UML/2.4.1/Infrastructure`
3. Ivers, J., et al.: Documenting Component and Connector Views with UML 2.0, Technical Report CMU/SEI-2004-TR-008, ESC-TR-2004-008. Software Engineering Institute, Carnegie Mellon (2004)
4. Niz, D.: Diagrams and Languages for Model-Based Software Engineering of Embedded Systems: UML and AADL, `http://www.sei.cmu.edu/library/reportspapers.cfm` (retrieved January 25, 2011)
5. Hnatkowska, B., Huzar, Z., Magott, J.: Consistency Checking in UML models. In: 4th Int. Conf. on Information Systems, Modeling ISM 2001 (2001)
6. Business Process Management Initiative, `http://www.bpmi.org`
7. van der Aalst, W.M.P., ter Hofstede, A.H., 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)
8. Organization for the Advancement of Structured Information Standards, `http://docs.oasis-open.org/wsbpel/2.0`
9. Workflow management Coalition, `http://www.wfmc.org/xpdl.html`
10. Object Management Group, Business Process Model and Notation, `http://www.bpmn.org`
11. Woodcock, J., Davies, J.: Using Z, `http://www.usingz.com/text/online/index.html`
12. Shroff, M., France, R.B.: Towards a formalization of UML class structures in Z. In: Proc. of the 21st Int. Computer Software and Applications Conf., August 11-15, p. 646 (1997)
13. Gero, J.S.: Design prototypes: a knowledge representation schema for design. AI Magazine 11(4), 26–36 (1990)
14. Truyen, F.: The Fast Guide to Model Driven Architecture, The Basics of Model Driven Architecture. Cephas Consulting Corp. (2006)
15. Spanoudakis, G., Zisman, A.: Inconsistency management in software engineering: Survey and open research issues. In: Chang, S.K. (ed.) Handbook of Software Engineering and Knowledge Engineering, vol. 1, pp. 329–380. World Scientific Publishing Co., London (1999)
16. Finkelstein, A., Gabbay, D., Hunter, A., Kramer, J., Nuseibeh, B.: Inconsistency Handling in Multi-Perspective Specifications. Transactions on Software Engineering 20(8), 569–578 (1994)

17. Egyed, A.: Instant consistency checking for the UML. In: ICSE, pp. 381–390 (2006)
18. Mens, T., Straeten, R.V.D., Simmonds, J.: A Framework for Managing Consistency of Evolving UML Models. In: Yang, H. (ed.) Software Evolution with UML and XML, ch. 1 (2005)
19. Usman, M., et al.: A Survey of Consistency Checking Techniques for UML Models. In: Advanced Software Engineering & Its Applications. IEEE (2008)
20. Kuzniarz, L., et al. (eds.) Workshop on "Consistency Problems in UML-based Software Development II" 2003, Research Report 2003:06, Blekinge Institute of Technology, San Francisco, USA (2003)
21. Choi, H., Yeom, K.: An Approach to Software Architecture Evaluation with the 4+1 View Model of Architecture. In: Ninth Asia-Pacific Software Engineering Conf., pp. 286–293. IEEE Computer Society (2002)
22. Kennaley, M.: The 3+1 Views of Architecture (in 3D): An Amplification of the 4+1 Viewpoint Framework. In: Seventh Working IEEE/IFIP Conference, pp. 299–302. IEEE Computer Society (2008)
23. Issa, A., Abu Rub, F.A.: Performing Early Feasibility Studies of Software Development Projects Using Business Process Models. In: Proc. of the World Congress on Engineering, WCE 2007, London, UK, July 2-4, vol. I (2007)
24. Dijkman, R.M., Joosten, S.M.: An Algorithm to Derive Use Case Diagrams from Business Process Models. In: 6th Intl. Conf. on Software Engineering and Applications (SEA), pp. 679–684. Acta Press, Anaheim (2002)
25. Odeh, M., Kamm, R.: Bridging the gap between business models and system models. Information and Software Technology 15(45), 1053–1060 (2003)
26. Kruchten, P.: The Rational Unified Process: An Introduction, 3rd edn. Addison-Wesley, Boston (2003)
27. Cabot, J., Gómez, C.: Deriving Operation Contracts from UML Class Diagrams. In: Engels, G., Opdyke, B., Schmidt, D.C., Weil, F. (eds.) MoDELS 2007. LNCS, vol. 4735, pp. 196–210. Springer, Heidelberg (2007)
28. Rasch, H., Wehrheim, H.: Checking Consistency in UML Diagrams: Classes and State Machines. In: Najm, E., Nestmann, U., Stevens, P. (eds.) FMOODS 2003. LNCS, vol. 2884, pp. 229–243. Springer, Heidelberg (2003)
29. Goel, A., Rugaber, S., Vattam, S.: Structure, behavior & function of complex systems: The SBF modelling language. Int. Journal of AI in Engineering Design, Analysis and Manufacturing 23, 23–35 (2009)
30. Bastarrica, M.C., Ochoa, S.F., Rossel, P.O.: Integrated Notation for Software Architecture Specifications. In: SCCC 2004, pp. 26–35 (2004)
31. Vondrak, I.: Business Process Modelling. In: Proc. of the 2007 Conf. on Information Modelling and Knowledge Bases XVIII (2007)
32. Elleuch, N., Khalfallah, A., Ahmed, S.B.: Software Architecture in Model Driven Architecture. In: 3rd Int. Symposium on Computational Intelligence and Intelligent Informatics – ISCIII 2007, Agadir, Morocco, March 28-30, pp. 219–223 (2007)
33. Lange, C., et al.: An Empirical Investigation in Quantifying Inconsistency and Incompleteness of UML Designs. In: Proc. Workshop on Consistency Problems in UML-based Software Development, 6th Int. Conf. on Unified Modelling Language (2003)
34. Wang, S., Jin, L., Jin, C.: Ontology Definition Meta-model based Consistency Checking of UML Models. In: Proceedings of the 10th Int. Conf. on Computer Supported Cooperative Work in Design. IEEE (2006) 1-4244-0165-8/06

Modeling Business Processes from Work Practices

Marielba Zacarias and Paula Ventura Martins

Research Center for Spatial and Organizational Dynamics
Algarve University, Gambelas, Faro - Portugal
{mzacaria,pventura}@ualg.pt

Abstract. Business process modeling methodologies need to pay attention to (1) the changing and distributed nature of business process, and (2) the contextual and tacit nature of the knowledge that operational actors have regarding business process. However, available methodologies offer little guidance to these concerns. This paper describes how to model business process models from work practices, using the BAM methodology. BAM is a methodology for business process modeling, supervision and improvement that works at two dimensions; the dimension of processes and the dimension of work practices. The paper illustrates BAM's business process discovery approach, which encompasses learning and modeling subphases, with a case study in an organizational setting.

Keywords: business process modeling, work practice modeling.

1 Introduction

Business process modeling (BPM) specializes on describing how activities interact and relate with each other, as well as their relationship with other business concepts such as goals and resources, where resources may be informational entities and human or automated actors. Business Process Modeling (BPM) methodologies are supported by data collection techniques including interviews, surveys, text/document analysis, among others. Nevertheless, it is argued that existing BPM methodologies are not appropriate to elicit the continuously evolving knowledge that is required in building business process models. Process-centric approaches tend to emphasize process (workflow, decision, information, activities) as the dominant dimension [1]. However, BPM would benefit from a better understanding of other elements that contribute to process execution such as interactions between activities, people, products, information and other resources. A further limitation stems from the tacit nature of process knowledge. Indeed, many organizations simply do not know their end-to-end processes accurately or in detail, since their process knowledge is tacit and decentralized [2]. Recent research in BPM is aiming to address the unpredictability of business processes [3,4], but there is yet little guidance in how to address the problem of tacit knowledge business process model maintenance.

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 440–454, 2012.

Work practice is a concept that originates in socio-technical systems, business anthropology, work systems design and management science [5]. Work practices are the behaviors expressed as action patterns of *specific* individuals, performing *specific* activities, in *specific* circumstances. Work practices involve people engaging in activities over time, not only with each other, but also with machines, tools, documents, and other artifacts. The importance of discovering work practices to improve user support has been acknowledged in [6,7]. Work practice modelling is also important to (1) address concerns disregarded by BPM such as providing a deeper understanding of the human activities composing business processes, and (2) assessing the alignment between process models and actual execution [8].

This paper describes how to model business processes from work practices using the Business Alignment Methodology (BAM). BAM is a multidisciplinary approach aimed at modeling, supervising and improving business process models, paying attention not only to process but to product, information and human dimensions through actual work practices. This approach is driven by three main ideas; first, that organizations need to model business process descriptions through a collaborative approach involving not only business analysts and process owners, but also all operational actors. Second, the dynamic nature of organizations and their environments results in frequently changing work practices that affect the pace of change in business processes. Third, BAM acknowledges the frequently tacit nature of the knowledge that operational use in executing business process knowledge.

The remainder of this paper is organized as follows; section 2 summarizes related work on business process modeling, work practice modeling and context modeling. Section 3 summarizes BAM methodology. Section 4 illustrates the first phase of the proposed methodology with a case study in a real organizational setting. Section 5 outlines our conclusions and outlook.

2 Related Work

2.1 Business Process Modeling

Several BPM frameworks propose describe the ways of building business process representations. Some approaches are systematic and detailed methodologies encompassing a set of procedures, techniques and tools to support the business process model construction. BPM methodological procedures involve of several steps and include descriptions of the inputs and outputs of each step [9,10,11]. Data collection techniques involve combinations of techniques drawn from the field of qualitative research, and include focused interviews, workshops and surveys. In some cases, templates to support data collection are also provided. These templates facilitate the recording of the model components (e.g. list of human and automated actors) and the relationship between them (e.g. actors of a given activity).

The semiotic-based approach developed by Dietz [12] provides a language-action perspective (LAP) methodology that guides the construction of business process models. Data sources are textual descriptions of the enterprise operation (no specific collection techniques are provided). Dietz conceptualizes operations at 3 levels; (1) performa (data), (2) informa (information) and (3) forma (transactions). Once collected, the descriptions are analyzed using two techniques (*performa-informa-forma* and *coordination-actors-production analysis*) that perform a semantic analysis of these descriptions. In the former, sentences corresponding to the *performa, informa* and *forma* levels are distinguished with different colors. The latter uses different types of brackets to distinguish `actors`, `coordination acts` and `production acts`.

2.2 Work Practice Modelling

Research efforts in work practice modelling aim at supporting system development. Pomerol and Brezillon developed a context model and representation language [13]. A premise of this work is that the main distinction between operational practices is the *context* where these practice apply. Their model of context relates the notion of context and knowledge. Sierhuis [7] propose an activity-based multi-agent modelling environment to model work practices. The authors develop a notion of situatedness supported by the following concepts; (1) people and knowledge, (2) situated action, (3) situated cognition, (4) situated learning and (5) autopoiesis. As a result, the representation language BRAHMS (Business Redesign Agent-based Holistic Modelling System) was developed to model knowledge in situated actions and learning in human activities.

These approaches regard the particularities of the agents performing activities and situations. Nonetheless, they were developed for systems development purposes. Consequently, the set of concepts provided require specialized skills and are thus too complex for the non-technical personnel typically involved in organizational analysis. Their linkage to other enterprise views is also not fully explored. An aspect not addressed in detail are the corresponding model building methods. Pomerol and Brezillon research focus more on work practice representation than the methods to capture it. Whereas the Brahms language is complemented with a systematic method to build the model, little detail is provided about how to *capture work practice.* Lundberg and Berquist [14] specifically focus this issue and describe a combined ethnographical approach, labelled by the authors as “eclectic ethnography”, to capture work practice for systems design.

2.3 Context Notions and Modelling Approaches

Work practice research shows the relevance of the notion of context. Nevertheless, despite several efforts to achieve a shared understanding around this notion, [15], the definition of context remains dependent on its application area. In cognitive sciences, B. Kokinov [16] developed a dynamic theory of context that defines it as the set of all entities that influence human (or system’s) behavior on a particular occasion. The main principles of the dynamic theory of context are: (1) context is a state of the mind, (2) context has no clear-cut boundaries, (3) context consists of all associatively relevant elements and (4) context is dynamic. Sociological approaches typically regard context as networks of entities (people, actors/agents and artifacts), that emerge from the interactions among them. Whereas some focus on the network elements, others focus on its emergent properties. In the latter case, context is regarded as sets of rules and resources that support and define interactions patterns among agents [17]. Activity Theory [18] and Actor-Network Theory [19] have been used in modelling social contexts.

In computer sciences, context is viewed as a collection of things (sentences, propositions, assumptions, properties, procedures, rules, facts, concepts, constraints, sentences, etc) associated to some specific situation (environment, domain, task, agents, interactions, conversations, etc). The artificial intelligence field has developed an extensive research on context. In a pioneer work, McCarthy [20] introduces contexts as abstract mathematical entities to allow axioms valid within limited contexts to be expanded to transcend its original limitations. In problem solving, Pomerol and Brézillon define context as the implicit constrains of each step of a problem. The notion of context developed by these authors is used in modeling problem solving practices at work. In human-computer interaction, Dey et al. [21] define context as any information that can be used to characterize the situation of entities. In context-aware computing, Dourish [22] draws from sociological approaches and proposes taking a different stance on the notion of context. Dourish argues that (1) rather than a kind of information, context is a relational property holding between objects or activities, (2) contextual features are defined dynamically, (3) context is specific to each activity or action, and (4) rather than separated from activity, context arises from the activity.

3 Business Alignment Methodology (BAM)

Research on business process modeling (section 2.1) and work practice modeling (section 2.2) are addressed separately and focus mostly on modeling languages. The novelty of BAM results from acknowledging and integrating work practice and business process views of business operations within a single methodological framework.

The BAM methodology (see figure 1) proposes a two-dimensional approach, encompassing three phases: (1) Business Process Discovery, (2) Business Process Supervision and (3) Business Process Assessment and Improvement. Business Process Discovery provides an initial process specification through interviews and collaborative methods. Business Process Supervision assures that daily practices follow base business process models. Business Process Assessment and Improvement allows analyzing performance measures to improve and refine business process models. Since the goal of this paper is to illustrate results from the Business Process Discovery phase and due to space limitations, this phase will be described in more detail than the remaining phases.

Each phase integrates two dimensions: (1) Process and (2) Practice. The Practice dimension explores day-to-day work based on individual and group actions and practices. This dimension captures and represents in various ways on-site information needed to systematically validate business process models. This dimension entails addressing the knowledge that operational actors (represented by individuals or groups), have of their own actions. In the practice dimension, knowledge is local and frequently tacit, thus it is hard to formalize. The Practice dimension covers information needed to systematically support or reject many process decisions based on the result of daily experiences. In the Process dimension, business analysts discover, review and improve business process descriptions, based on information of the Practice dimension. The process dimension addresses knowledge that crosses functional divisions and organizational boundaries (clients, suppliers). Therefore, organizational processes embody specific accumulated knowledge that is not confined to particular individuals or groups. This knowledge, which is explicit, needs to be transmitted and shared among the individuals and groups working at the functional divisions responsible for executing the corresponding business process. The process dimension also addresses the need of continuous business process supervision and improvement as a reaction to fast-changing environments in the business world. These two dimensions, Practice and Process, will ensure the proper structure to articulate individual, group and organizational knowledge with the knowledge of business analysts.

3.1 Business Process Discovery

The main goal of a Business Process Discovery (BPD) is to get personal descriptions of business processes from work practice descriptions. BPD phase aims at developing an organizational profile of people, activities, technology, and information in order to capture actual business processes. This phase includes two main subphases: (1) Learning (Eliciting) Business (LB) and (2) Modeling Business (MB).

Learning Business. The Learning Business subphase encompasses three activities; (1) kickoff meeting, (2) eliciting information of practice and process, and (3) elaborating preliminary practice and process descriptions. The kickoff meeting communicates operational actors the goal and procedures of the BPD phase. Information elicitation is accomplished according to the nature of each dimension. The outcome of this subphase are preliminary descriptions of both work practices and business processes.

Practice Dimension. In our framework, work practices are defined in terms of action patterns, that is, recurrent action sequences. Due to its local nature, work practices vary according the context of execution. Moreover, operational actors are often unaware of their recurrent action patterns. Hence, instead of conducting standard interviews and workshops, the approach to elicit work practices is

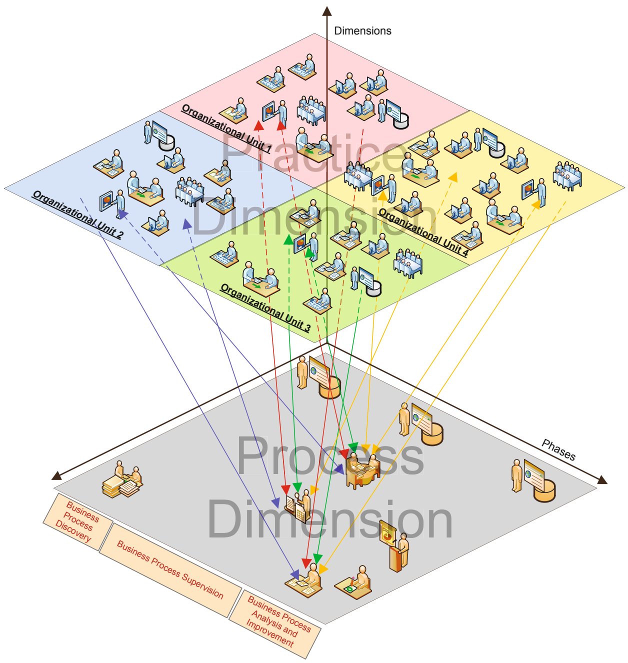


Fig. 1. BAM overview

accomplished as follows: (1) Capturing daily actions, (2) Identifying action and interaction contexts created by related actions, and (3) discovering recurrent action patterns within contexts.

Capturing daily actions creates action repositories, where each action is registered as <actor, action, resource> triples. Actions refer to fined-grained operations of actors' daily work. Actions are identified with verbs taken from the vocabulary shared among operational actors. Resources may involve information, tools, materials or even human knowledge not yet externalized in external sources. Resources are described with nouns or nominal phrases using actors' own vocabulary. Actions may be communicative or not communicative. Communicative actions involves two actors; a sender and a receiver.

Understanding the meaning of actions requires situating them in a particular context. Drawing on the sociological perspective (section 2.3), we regard contexts as situations resulting from the interactions between an actor and resources or between different actors. Such interactions are reflected in actions. **Action contexts** are defined as situations created by action streams performed by one or more operational actors. Action streams performed by a single individual create **personal action contexts**. Interaction streams i.e. communicative actions exchanged between two or more actors that are part of a single conversation, create **interaction contexts**. Under this definition, action and interaction contexts are uncovered by grouping sequences of actions related to a given situation. Once identified, contexts are analyzed in order to discover recurrent action patterns within them. It is noteworthy that action and interaction contexts and patterns are not generic. Rather they refer to specific persons, places a time periods.

Process Dimension. The action patterns discovered at the practice dimension are then analyzed and discussed by operational actors and business analysts in order to define business processes, as well as the business activities and resources, composing business processes. This discussion entails an aggregation process that is accomplished in a bottom-up fashion. However, a top-down application of high-level knowledge of the organization such as organizational goals and strategies is required in driving the definition of business process. The dynamic interplay between these two dimensions (practice and process dimensions) shows the synergy between key operational actors and activities described by business analysts involved in BPD.

Learning subphase roles. The Learning Business subphase involves two roles; (1) operational actors are responsible for registering daily actions and identifying action and interaction contexts and (2) business analysts who are responsible for conducting the kickoff meeting, discovering action patterns, and associating them to business activities, roles, and resources.

Modeling Business. The subphase Modeling Business, involves several stakeholders (business analyst, process owner, organizational unit responsible and operational actors) that perform three interrelated activities (1) model construction; (2) model revision and evaluation and negotiation and (3) model approval. These activities support a negotiation process that if successful, results in a shared view of the process. Finally, the model approval activity concludes the interaction process and collaboration among the parties involved in a business process model specification by approving or rejecting the model. The techniques used in model construction vary according the dimension.

Practice Dimension. The action and patterns identified within particular contexts in the previous subphase are shared, discussed among operational actors and business analysts involved in similar activities in order to identify which practices yield better results.

Process Dimension. The process representation concerns activities, resources, decision points and work flows (topology). In the Process dimension, business analysts use the best practices that lead to business process reviews and improvement. Action patterns represent alternative proposals that result from the execution of different individuals and groups who may act based on different assumptions and meanings. Business analysts then define business process models based on previously identified best practices.

Modeling subphase roles. The Modeling Business subphase involves four roles; (1) operational actors are responsible for assessing their practices and identifying best practices, (2) organizational unit responsible is responsible for modeling work practices (3) business analysts are responsible for helping operational actors in identifying best practices, and helping the organizational unit responsible in modeling work practices and (4) process owners are responsible for building process models based on best practices.

3.2 Business Process Supervision

In the Business Process Supervision (BPS) phase, formal control mechanisms are designed in order to ensure that operational actors carried out real business activities as described by business models. Control mechanisms consist of two main activities: (1) compare work practices with base business process models, and (2) identify new business process model descriptions. The outcome of this phase are revised versions of base business process models. This phase ends when business analysts and operational actors agree that: business process model describe the detailed behavior that address real needs, major problems have been solved, business process practices provides some useful value to the organization and these practices are stable enough to implement a new and improved business process version.

3.3 Business Process Assessment and Improvement

In this phase, the business analyst analyze change proposals and through a comparison between base business process models and proposed changes, a new set of models is build to correct work that is not proceeding well, by showing where adjustments need to be made. In the end, the results gathered during assessments enable improvements and consistent refinements in order to produce an improved set of business process models. This phase ends when all the involved actors agree that the objectives set during BPD (and modified throughout the second phase) have been met; especially if all participants are satisfied with the new business process model version.

4 Case Study

4.1 Organizational Setting

The Business Process Discovery phase of the methodology was tested by a software development team of 4 programmers (ages between 25 and 32), and the project leader (age 35), who performs both programming and project management tasks. The team develops web applications for a commercial bank. Team members perform systems analysis, design, programming, test and maintenance activities. During the observation period, the team worked on the following applications; (1) Suppliers, (2) Claims, (3) Client correspondence management (called Mail application), (4) Evictions and (5) Marketing Campaigns. Being a key user and a small case, the team manager worked also as the business analyst. The team manager's chief played the process owner role.

4.2 Business Process Discovery Results

The research goal and data collection methods were discussed in a briefing session. Worksheets with templates to registered actions were distributed. In order to achieve some standardization regarding the terms used, the meeting also served to discuss typical action names and resources.

nº	day	Actor/Sender	Receiver(s)	action type	description	tools	documents| information	human
1	6	Carla	team members	propose	a team meeting at 15h	e-mail	team member addresses	
2	6	Pproduction area	Mariana, Catarina	inform	error in automatic table updates	e-mail	error message	
3	6	Carla	Catarina	command	find a solution to the automatic table update problem	e-mail	Catarina' address	
4	6	Carla	CG team	propose	test claims application (integration tests)	telephone	CG team tlf. number	
5	6	CG team	Mariana	accept	test claims application (integration tests)			
6	6	Carla		prepare	prepare claims application test environment	claims application	test data	claims application knowledge (testing procedure)

Fig. 2. Some registered actions

Learning phase - Practice dimension. A set of 534 actions was manually collected through a three-week observation period. Figure 2 shows an extract of an action log that illustrates the structure defined for actions. Due to human multitasking, grouping actions in personal contexts is essential to distinguish related from unrelated actions. This separation is made through the notion of personal contexts. Personal contexts are discovered by grouping together action sequences performed by a given individual and belonging to a given situation or topic. Such groupings allow defining context features such as frequent action types and resources, and labeling each context.

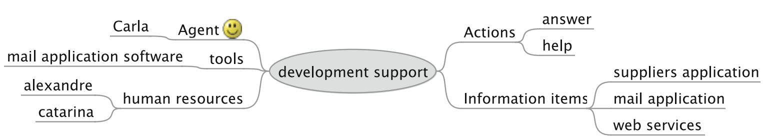


Fig. 3. Carla's personal context "Development Support"

Figure 3 depicts the context features of the personal contexts of the participant subject Carla, labeled as "Development Support". This context is characterized by the keywords *answer, help* (actions), *suppliers application, mail application, web services* (information items), *mail application software* (tools); and

Subject	Context ID	Context Name
alexandre	*a3*	Evictions Web Service Problem
alexandre	*a4*	Team Meetings
carla	*c2*	Development support
carla	*c3*	Team Meetings
catarina	*t3*	Discussions/Collaborations
goncalo	*g2*	Discussions/Collaboration with Catarina
goncalo	*g3*	Development and User Support
mariana	*m1*	Project Management Reports and Meetings
mariana	*m10*	Campaings Application Adjustments
mariana	*m3*	Integration Tests
mariana	*m4*	User Support
mariana	*m51*	File Upload Component
mariana	*m6*	Evictions Web Service Problem
mariana	*m7*	Script Execution Problem
mariana	*m8*	Suppliers Application
mariana	*m81*	Web Components Programming

Fig. 4. Some Personal Contexts

alexandre, catarina (human resources). Figure 4 shows some personal contexts identified for the participants of our case study.

Nonetheless, to properly understand execution, it is not enough to model personal contexts and individual behaviors. Most tasks are executed by several individuals. Hence, it is necessary to identify and characterize interaction contexts. The analysis of interaction contexts allows to see which resources are shared (and how) between different individuals.

Day	Nº	Ag-Sender	Contex	Ag-Receiver	Action	Emb. Action	Description Item Keywords
9	111	alexandre	a3	mariana	request	publication	quality environment
9	111	alexandre	a3	mariana	request	publication	updated evictions web service
9	115	mariana	m6	alexandre	request	test	quality environment
17	315	alexandre	a3	mariana	request	publication	evictions web service
17	315	alexandre	a3	mariana	request	publication	quality environment
17	318	mariana	m6	alexandre	inform	publication	evictions web service
17	318	mariana	m6	alexandre	inform	publication	quality environment
22	393	mariana	m6	alexandre	request	discuss	evictions web service
22	402	alexandre	a3	mariana	inform	NULL	evictions web service

Fig. 5. Evictions Web Service Action Stream

Whereas personal contexts are identified from action streams of a single individual, the identification of interaction contexts is made from action streams from *two* or more given individuals. Interpersonal contexts relate two specific personal contexts of interacting individuals. Figure 5 depicts an action stream created by a problem detected on the "evictions web service". This action stream involves several interactions between Alexandre's personal action context `a3`, and Mariana's personal action context `m6`. Interactions involve two actors; a sender and a receiver. Drawing from Dietz's work [12], interactions involve in practice two actions; a coordination act (labeled as action) and a production act (labeled as embedded action). The coordination act entails the execution of a production act. The production act has associated resources (labeled as description item keywords). Table 1 shows some interaction contexts, and their associated personal contexts. Notice that the contexts ending with an x, belong to individuals or groups external to our participant subjects.

Table 1. Some interaction contexts

Interaction Context	Personal Contexts	Description
ic5	$< c2 - m8 >$	suppliers app. support
ic6	$< g2 - t3 >$	suppliers app. support
ic7	$< g3 - m8/m81 >$	suppliers app. support
ic8	$< m1 - a4/c3/g5/t6 >$	team meetings
ic9	$< m1 - antx >$	project management reports
ic11	$< m3 - cgTeamx >$	integration tests
ic16	$< m4/m51/m6 - pubTeamx >$	software publication
	$< m7/m8/m10 - pubTeamx >$	

Identifying interaction contexts allows uncovering actual action patterns used in executing activities. Hence, interaction contexts allow assessing how these patterns differ among different groups and from pre-defined business process models. Table 2 depicts the action patterns identified within the action repository of this case study. With the data collected, in this case it was possible to identify recurrent action sequences composed of two to six action types. Actions in italic represent actions that do not appear within all sequences but that were inferred, based on the fact that they need to be executed in order for the remaining actions to take place.

Table 2. Some action patterns

ID	Context Name	Action Pattern
ic5	suppliers application support	1.request (help) - 2.help
ic6	suppliers application support	1.ask - 2.answer
ic7	suppliers application support	1.request (help) - 2.help
		1.request (solve) - 2.solve
ic8	team meetings	1.propose - 2.accept - 3.assist
ic9	project management reports	1.request (update) - 2.update-3.send
ic11	integration tests	1.request (test) - 2.test
		1. inform(test results) - 2. test
ic16	software publication	1.request (publication) - 2.perform (publication)
		3.test - 4.inform (publication)
		5.inform (publication)-*6.inform(publication)*

Learning subphase - Process dimension. Action patterns uncovered at the Practice Dimension can be used in a bottom-up fashion to discover actual business activities, processes, and resources. However, in this case, the organization had previously accomplished a business process modeling initiative, having already identified a list of pre-defined activities composing business processes. Hence, in this case, action patterns were associated to those activities. Due to space limitations, we only mention the pre-defined activities related to the action patterns depicted in table 2: *(1) Test application components (ic11), (2) Publish application components (ic16), (3) Support users (ic5, ic6, ic7), (4) Ellaborate project reports (ic9), and (5) Monitor Project Progress (ic8).*

Action repositories include descriptions of the information items, tools, materials or knowledge used or produced by each action. However, at the process dimension those items need to be linked with formally defined resources already associated to given business activities or processes. In this case, a list of such resources was pre-defined together with business activities. The table depicted in figure 6 shows the relationships between most frequently found items and activity resources.

Description Item	Freq.	Formal Resource
team meeting	52	projects
quality environment	33	application status
suppliers application	32	suppliers application
suppliers application web component	28	suppliers application
mail application	28	mail application
suppliers application data management class	24	suppliers application
common services application	23	common services application
suppliers application application class	22	suppliers application

Fig. 6. Associating action resources to activity resources

Modeling subphase - Practice dimension. As aforementioned, the action patterns found in specific contexts allows uncovering action patterns of specific individuals or groups. For example, figure 7 depicts a publication practice of some team members. This practice was uncovered from the action pattern found in interaction context ic16. Notice how practices are specific to particular persons and tools used by them. It is also noteworthy that a practice maybe related to more than one formally defined activity. The publication practice includes actions belonging to two pre-defined business activities; (1) test and (2) publish software components. It is important to notice that action repositories were frequently incomplete. However, the information they provided was enough to "fill in the blanks". Actions depicted in solid-line boxes are actions present in the repository. Traced-line boxes represent actions that were not always present in the repository, but operational actors knew they were performed always. Dotted-line boxes are actions that although were never registered, operational actors acknowledged them as actions that were executed in between two given registered actions.

Modeling subphase - Process Dimension. After collecting diagrams representing several practices, the team discussed them and selected the best practices. Best practices were then used to build a software development process to be shared by

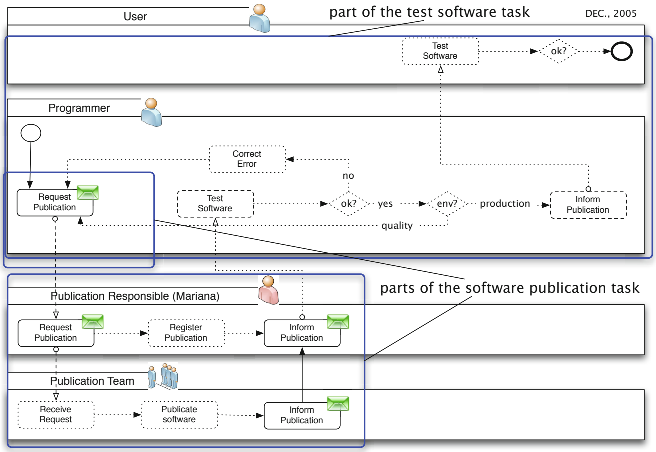


Fig. 7. Publication practice

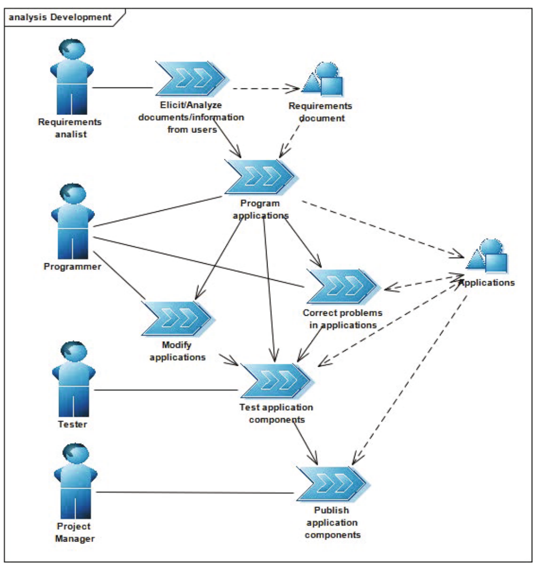


Fig. 8. Generic Software Development Process

all teams. Figure 8 depicts a process model resulting from this phase. The figure shows the process that resulted from putting together the best practices related to software development, or from redefining current practices. The resources shown in the figure were uncovered as illustrated in figure 6.

Case Study Evaluation. This phase of BAM was qualitatively evaluated through a non-structured interview with the team members. From this interview, we conclude that: (1) though the information acquired from the action repository was incomplete, missing actions were easily acknowledged to 'filled in' the representations, (2) whereas registering actions manually restricted the number of actions registered and registration period, this effort was partially compensated since it minimized the need of meetings and external observers, (4) practice and processes diagrams provided a common ground in comparing practices among teams, and with formal process models.

5 Conclusions

This paper illustrates an application of the BAM methodology. Driven by the decentralized, tacit and dynamic nature of business processes, BAM's design is structured in three phases and two dimensions. This paper illustrates the Business Discovery phase for its both dimensions i.e Practice and Process, through a case study conducted in a real organizational setting. Case evaluation was conducted through interviews with all participants, where operational actors found easy identifying actions and reaching agreements regarding their work practices since their meaning was very close to their everyday work. All operational actors found the action repository was very rich and detailed, where missing actions could be inferred to fill in practice representations, including practices they were previously unaware of. Both the team manager (business analyst) and the supervisor (process owner) indicated that having work practice descriptions was very helpful in discovering business processes.

The paper offers a partial illustration of the methodology. More extensive case studies encompassing the whole methodology, provided with formal evaluation techniques are required in order to have a more comprehensive evaluation of BAM. An exploration and development of automated methods for data collection and analysis are also essential in order to enable larger and longer case studies.

Acknowledgments. This article was partially financed by Fundação para a Ciência e a Tecnologia.

References

1. Hollingaworth, D., Services, F.: The Workflow Reference Model: 10 Years On. In: The Workflow Reference Model: 10 Years On, pp. 295–312 (2004)
2. Verner, L.: BPM: The promise and the challenge. Queue 2(1), 82–91 (2004)
3. Reichert, M., Dadam, P., Jurisch, M., Kreher, U., Goser, K.: Architectural design of flexible process management technology. In: PRIMIUM Subconference at the Multikonferenz Wirtschaftsinformatik (MKWI), Garching, Germany (2008)
4. Mutschler, B., Weber, B., Reichert, M.: Workflow management versus case handling: results from a controlled software experiment. In: Proceedings of the 2008 ACM Symposium on Applied Computing, Fortaleza, Ceara, pp. 82–89. ACM (2008)

5. Sierhuis, M., Clancey, W.J., Hoof, R.V., Hoog, R.D.: Modeling and simulating work practices from apollo12. In: 6th International Workshop on Simulation for European Space Programmes. ESTEC (2000)
6. Brézillon, P.: Using context for supporting users efficiently. In: 36th Annual Hawaii International Conference on System Sciences (HICSS 2003), p. 9. IEEE - Computer Society Digital Library (2003)
7. Sierhuis, M., Clancey, W.J.: Knowledge, practice, activities and people. In: AAAI Spring Symposium on Artificial Intelligence in Knowledge Management. Spring Symposium Series Technical Reports, pp. 142–148. AAAI Press (1997)
8. Zacarias, M., Serendero, P., Pinto, H.S., Tribolet, J.: Capturing and modeling work practice: A context-based approach. Revue d'Intelligence Artificialle 5(22), 669–688 (2008)
9. The Open Group: The Open Group architectural framework (TOGAF) - version 8.1 (2003)
10. Spewak, S.H., Hill, S.C.: Enterprise Architecture Planning: Developing a BluePrint for Data, Applications and Technology. John Wiley & Sons (1992)
11. CAP Gemini: Integrated architecture framework IAF (2007)
12. Dietz, J.L.G.: Enterprise Ontology. Springer (2006)
13. Pomerol, J.-C., Brézillon, P.: About Some Relationships Between Knowledge and Context. In: Akman, V., Bouquet, P., Thomason, R.H., Young, R.A. (eds.) CONTEXT 2001. LNCS (LNAI), vol. 2116, pp. 461–464. Springer, Heidelberg (2001)
14. Lundberg, N., Bergquist, M.: Capturing work practice: Applying combined ethnographical approaches in field studies. In: 23rd Information Systems Research Seminar In Scandinavia, IRIS (2000)
15. Bazire, M., Brézillon, P.: Understanding Context Before Using It. In: Dey, A.K., Kokinov, B., Leake, D.B., Turner, R. (eds.) CONTEXT 2005. LNCS (LNAI), vol. 3554, pp. 29–40. Springer, Heidelberg (2005)
16. Kokinov, B.: A dynamic approach to context modeling. In: Brezillon, P., Abu-Hakima, S. (eds.) Working Notes of the IJCAI 1995 Workshop on Modelling Context in Knowledge Representation and Working Notes of the IJCAI 1995 Workshop on Modelling Context in Knowledge Representation and Reasoning, pp. 59–72 (1995)
17. Giddens, A.: The Constitution of Society. University of California Press (1984)
18. Kaptelinin, V., Nardi, B., Foot, K. (eds.): Acting with Technology: Activity Theory and Interaction Design. MIT Press (2006)
19. Latour, B.: Reassembling the Social: An Introduction to Actor-Network. Oxford University Press (2005)
20. McCarthy, J.: Notes on formalizing context. In: International Joint Conferences on Artificial Intelligence, IJCAI 1993 (1993)
21. Dey, A., Abowd, G., Salber, D.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Human-Computer Interaction 16(2-4), 97–166
22. Dourish, P.: What we talk about when we talk about context. Personal and Ubiquitous Computing 8(1), 19–30 (2004)

Experimentation in Executable Enterprise Architecture Models

Laura Manzur, John Santa, Mario Sánchez, and Jorge Villalobos

Universidad de los Andes, Bogotá, Colombia
{lc.manzur441,j-santa,mar-san1,jvillalo}@uniandes.edu.co

Abstract. Enterprise Architecture (EA) is a multidimensional model-based approach which enables analysis and decision-making in organizations. Currently, most EA approaches produce inherently static models: they focus on structural qualities of the organizations and represent their state only in one specific point in time. Thus, these models are not suitable enough for analyzing dynamic and run-time features of the organizations. This paper aims to solve this situation by proposing a model-driven platform for EA modeling and simulation. The proposal includes the means to build executable EA models, define experiments over the models, run the experiments, observe their run-time behavior, and calculate indicator-based results to aid the decision-making process.

Keywords: Enterprise Architecture, Executable Models, Discrete Event Simulation.

1 Introduction

The core of Enterprise Architecture (EA) approaches are models and diagrammatic descriptions of systems and their environment [7]. The main goal of EA projects is building said models, ensuring that they are an adequate representation of the enterprise in a specific point in time, and to use the models to diagnose problems or to propose improvements.

Usually, EA projects use metamodels to guide the construction of EA models. These metamodels are selected or designed depending on the project's needs and they define the types of elements that can appear in the models, the relevant structural and behavioral properties of each one, and their possible relationships. For example, a metamodel can establish: *i*) that the sole entities supported are Processes, Applications and Infrastructure Services; *ii*) the structural and behavioral properties that modelers can provide about each kind of entity (e.g., application name and service cost (structural), application response time and application failure probability (behavioral)); *iii*) that processes and applications can be related (a process supported by an application), and that applications and infrastructure services can be related as well (an application depends on some services). Furthermore, some metamodels include *constraints*, which provide additional information about valid or desirable model structures. Constraints can be used to specify that the total cost of the services invoked in a process instance

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 455–469, 2012.

should be less than $10, or specify that no process should depend on more than three applications (desirable).

EA models are typically analyzed by human experts. They study the elements, relations, and properties to identify problems and to derive conclusions that support decision-making processes. These analyses are made both by hand (studying the models directly) and by using tools that interpret the information present in the models. Nevertheless, "an enterprise architecture [...] is a static description of the essential components of the enterprise and their interconnections. By itself, this static description does not provide enough information to analyze and understand the behaviors that a given enterprise architecture is capable of producing" [6]. This means that there is information about the enterprises, and especially about their behavior, that cannot be evidenced just by looking at the models. As a result, this information is ignored during the analysis processes.

Including dynamic elements in enterprise architecture models is not straightforward. On one hand, dynamic elements are typically absent from standardized metamodels, and thus they cannot appear in the models. On the other hand, analysis tools, which are based on said standardized metamodels, provide only the means to work with structural and static aspects. Thus, they would not be able to use information about dynamic features even if it would be present. Finally, dynamic elements introduce a whole new complexity level for the analysis process that is beyond the capabilities of some analysis techniques.

To address this situation and enable the modeling and analysis of dynamic features, we propose a *model-driven platform for simulation*. Simulation is the key element here: by means of simulation, it is possible to draw inferences concerning run-time characteristics of an enterprise [1]. Moreover, our proposal has some characteristics that differentiate it from previous simulation proposals in the EA domain. First of all, it does not use pre-defined metamodels to describe the EA models that are simulated. Instead, specialized metamodels can be defined by metamodelers depending on their own particular needs, and on the kind and complexity of the analysis that they wish to perform. Secondly, the platform supports arbitrarily complex behavior without requiring changes to the base platform (e.g., simulated applications can perform parallel operations, organized by priority queues, and with different response times depending on their complexity). Thirdly, simulation results are processed offline: runs are observed, events and intermediate results are traced, and metrics are calculated. Metrics are defined by EA analysts, and they correspond to the indicators needed for their analysis and decision-making processes. Finally, this simulation approach uses highly configurable scenarios, that can be reused across many experiments, making the platform very suitable for answering *what if* questions.

This paper is structured as follows. Section 2 introduces the simulation metamodel and EA metamodel, which are the foundation of the simulation platform, and section 3 explains how these are used to build simulation scenarios together with indicators. Section 4 discusses simulation experiments, and section 5 presents and ideal workflow to use the proposed platform, as well as the results of an experiment. Finally, section 6 discusses related work and section 7 concludes the paper.

2 Metamodels for EA Simulations

Our simulation approach is based on two kinds of metamodels. The first one, represented by a generic *Simulation Metamodel* (SMM), abstracts concepts common to every simulation project, as well as their execution logic. Figure 1 presents the elements of the SMM. The central element is *Controller*, which maintains a timeline consisting of discrete *Instants*. An *Instant* is a point in time in the future[1] on which *Future Happenings* are scheduled to occur. *Future Happenings* represent meaningful events generated by the execution of the simulation, such as the completion of an Activity, Task or Operation; or the failure of an Application. A happening is described by an action that will happen to a *Referenced Element*, which is any of those defined in the Simulation Model. A special type of happening we called *Future Stimulus* represents events defined as input to the simulation (e.g., a client initiating a business process). In this case, the referenced element must be an instance of an element type that inherits from the *Stimulable* type. Section 4 presents further detail about stimuli.

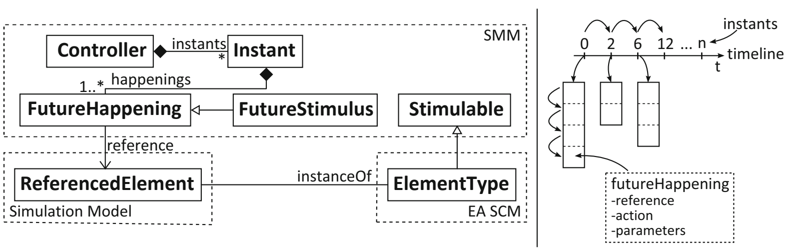


Fig. 1. Structure of SMM and timeline

During a simulation execution, the controller processes instants in chronological order. When all the happenings in an instant have been executed, the controller advances to the next one. To run the execution, we leverage on Cumbia [10][11]. Cumbia is a meta-modeling platform that supports the execution of models following the behavioral semantics specified in the metamodels, as well as complex monitoring requirements. In Cumbia, the elements of a metamodel are all *open objects*, which is a modeling abstraction formed by an *entity*, a *state machine*, and a set of *actions*. Since our proposal uses Cumbia as execution platform, the elements of the SMM must be described as open objects. For further information on Cumbia, refer to [11].

[1] The future is relative to the simulation execution. It refers to a point in time further from the current executing instant.

The second kind of metamodel in our approach defines the specific element types that will appear in the simulation models. On top of the typical characteristics of EA metamodels (attributes and relationships), these metamodels also include information about the behavior of each element type. There are certain concepts that appear only during the execution of a simulation such as '*process instance*', '*service invocation*', and '*current employee tasks*', which are not to be found in traditional EA metamodels. Concepts like these must be taken into account if we want the simulation to be as close to reality as possible, since they intervene in the operation of the real enterprise. This is critical in our approach, and requires this behavioral information to be detailed enough to allow the execution of the models. The advantage of this approach revolves around the ability to define behavior with arbitrary granularity and complexity, thus customizing the level of detail of the simulation. We call these metamodels, which are built on top of the SMM, *Simulation Capable Metamodels* (SCM).

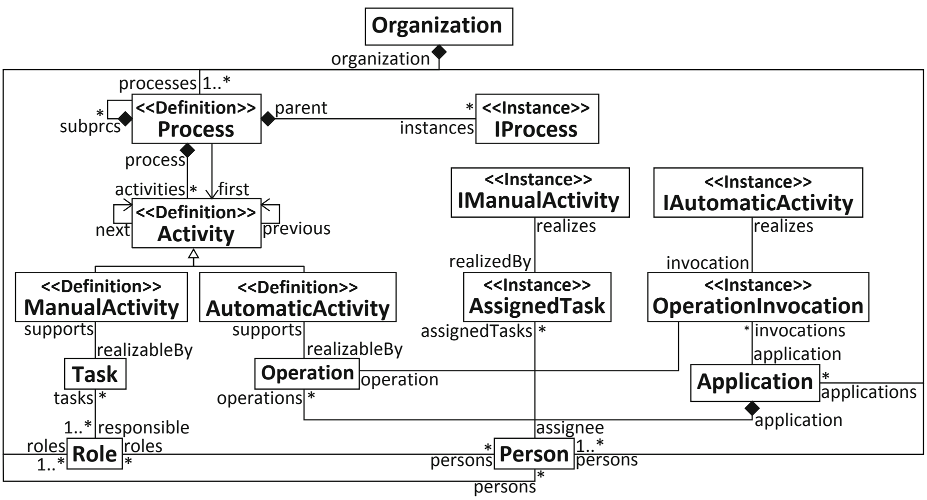


Fig. 2. EA SCM example

Figure 2 presents a simplified version of a SCM[2] from our metamodel repository. This SCM includes three EA domains: processes, applications and people (human resources). It models an *Organization* composed by a set of *Roles*, *Persons*, *Processes*, and *Applications*. Two types of activities are allowed: *ManualActivities* and *AutomaticActivities*, which are supported by *Tasks* and *Operations* respectively. Additionally, ⟨⟨*definitions*⟩⟩ are presented separately from ⟨⟨*instances*⟩⟩ because the behaviors of these kinds of elements are significantly

[2] Some relations between elements have been removed for legibility. The relations that present no multiplicity, correspond to 1. For a complete version of the metamodel, refer to [13].

different: Definitions are elements that provide a structure for guiding the creation of new instances; Instances are the elements that are executed and take time to be completed. This SCM also includes concepts and relations that appear only during the execution of the simulation, such as *invocations* (from Application to *OperationInvocation*) and *assignedTasks* (from Person to *AssignedTask*).

Because of the use of Cumbia as execution platform, the behavior of the elements is defined by state machines. These state machines define possible execution states for each element, and define the actions to perform during a simulation run. State machines are coordinated with events that trigger transitions; and with the invocation of methods implemented in the entities of the open objects.

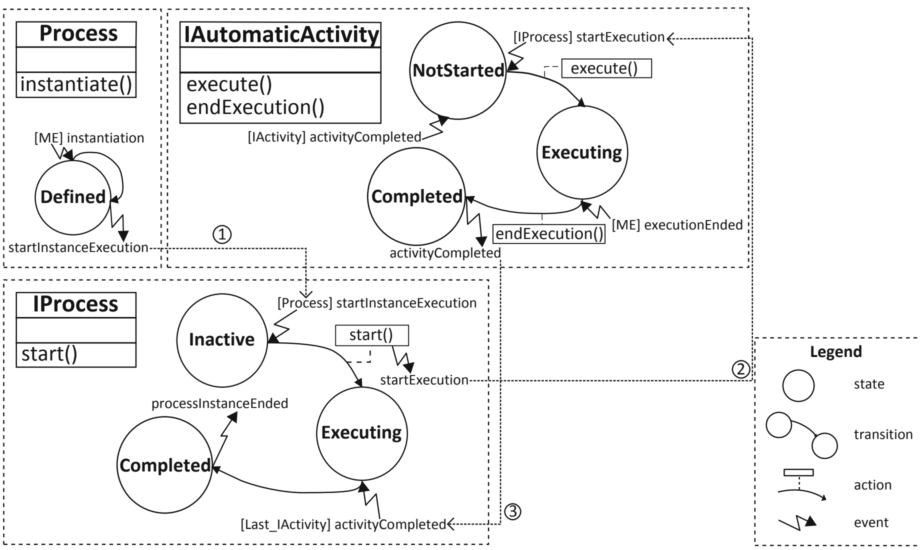


Fig. 3. State machines example

Figure 3 shows the open objects for elements Process, IProcess and IAutomaticActivity, and the coordination between them. It is necessary that every element of the SCM defines its corresponding state machine. Due to a lack of space, we will only explain the state machines from the mentioned open objects. A Process has only one state and an operation that creates an IProcess (`instantiate()`). IProcess and IAutomaticActivity include states that are coherent with the states of a process instance and an activity instance during their execution. When a process is instantiated during a simulation run, the Process element produces a `startInstanceExecution` event which is detected by the IProcess state machine (arrow 1), causing it to take the transition to the `Executing` state. This transition has a `start()` operation associated that produces a `startExecution` event. This event triggers the execution of the state machine of the first activity of the process. For this particular example, it is the IAutomaticActivity state machine (arrow 2). Once an activity has finished executing (transitions from `Executing` to `Completed`), it generates an `activityCompleted` event, which triggers the initiation of the next activity. Finally, when the last activity ends, it produces an `activityCompleted` event which notifies its parent IProcess (arrow 3), making the state machine take the transition to the `Completed` state.

The element IAutomaticActivity is a good example of the capabilities of the approach. Currently, the operation `execute()` finishes after an amount of time that depends on the current workload of the Application that realizes the operation. The calculation of the time is based on a probability distribution. The relations between element types of the architecture and how they behave can be easily modified by extending and adapting the metamodel [10].

To increase the reusability of the simulation models that conform to a given SCM, it is possible to set the values to any of its element's attributes from a configuration file that is external to the model. In this way, distinct configurations of the same simulation scenario can be tested without requiring new models for each case. We call this capacity parameterizability and models with this characteristic, *Parametric Simulation Models* (PSM).

3 Simulation Scenarios

A simulation scenario represents a simplified version of an organization which is interesting to be simulated, observed, and used to support decision making processes. Typically, one scenario is simulated several times with minor configuration differences to see which one offers the best outcomes. Moreover, to have comparable experiments, the characteristics observed in each one must be the same. To support this, a simulation scenario is composed by a PSM based on an SCM, and a set of *indicators* to group and present the results of simulation runs.

Figure 4 shows a PSM that conforms to the SCM described in section 2, which represents a small part of *Banco de los Alpes* (BDLA). BDLA is an EA scenario in the banking domain, which has been developed in our research group with the collaboration of representatives from different real banks, as a test case for several research initiatives related to EA [12].

The model represents the process for searching potential clients for the bank (*SearchNewClients*) which is composed of four activities: *LoadProspectsFromPartners*, *FilterUndesiredProspects*, *SegmentProspects* and *NotifyApprovedProductsToProspects*. These activities are respectively supported by four applications: an *FTP-ETL*, the *Clinton-MoneyLaunderingList*, a *CRM* and *ERP*. Notice that some element types do not appear in the PSM. This is because during the execution of this model, elements from concepts like *IProcess* and *OperationInvocation* will be created to represent instances of processes produced along the simulation execution and invocations to operations that support the process' activities. Therefore, elements of any type defined at the SCM with the Instance stereotype must not be defined in the PSM, but appear as the result of the simulation run.

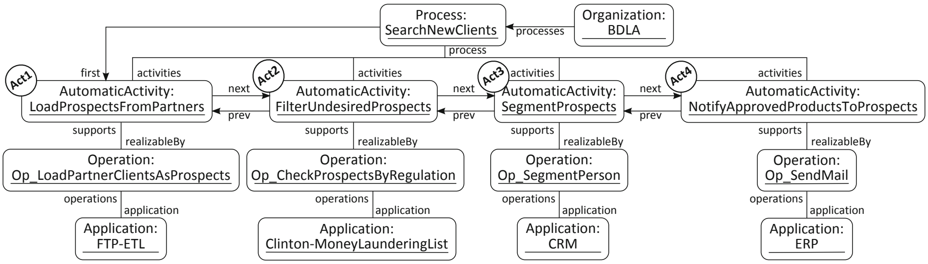


Fig. 4. PSM example

To complete a simulation scenario, a set of *Indicators* is defined to guide the collection of data about the simulation. As expressed by Frank in [5], an indicator refers to "quantitative measures that are specified using several mathematical constructs [...] for different types of reference objects and on various levels of abstractions in the enterprise". In our case, indicators refer to the specification of measurements to collect during the simulation. Indicators are described using an *Indicator Language*, which specifies how indicators should be presented and how to calculate them. Due to a lack of space to describe every aspect of our solution, we cannot give much detail about this language. Nevertheless, we will briefly present an example.

To calculate indicators, it is necessary to *observe* the simulation execution, by instrumenting the OpenObjects to gather relevant information. We took inspiration from the work in [9] to build an *Observation Structure*, composed by *Sensors* and *Tracers*, to collect data during the simulation execution. Sensors are in charge of monitoring elements in the simulation models, and detecting state changes in their state machines. When these occur, a sensor collects data and passes it to tracers. Tracers create *traces* with said data, and store them for offline processing. An observation model, which is automatically derived from an indicators definition, specifies the model elements to monitor, their events/actions of interest, the information that will be stored, and the corresponding tracers.

Figure 5 shows how this works. Firstly, a *Business Analyst* defines indicators (Is) by specifying (1) the desired visualizations of the simulation results, and (2) how to calculate them using variables (V). Additionally, he specifies the element types from the metamodel whose instances [from the model] (E) need to be monitored. The selection of these element types includes the places where sensors must be placed in the state machines. These places are called observation points. The sensors placed in these observation points are associated to a tracer (Trc) where traces (T) of intermediate states of the simulation are stored. After a simulation run is finalized, traces are processed and consolidated into indicators' representations, which are then presented to *Business Analysts* through the *Visualization Layer*. The indicator language's design is based in the metamodel proposed by Frank et al. in [5].

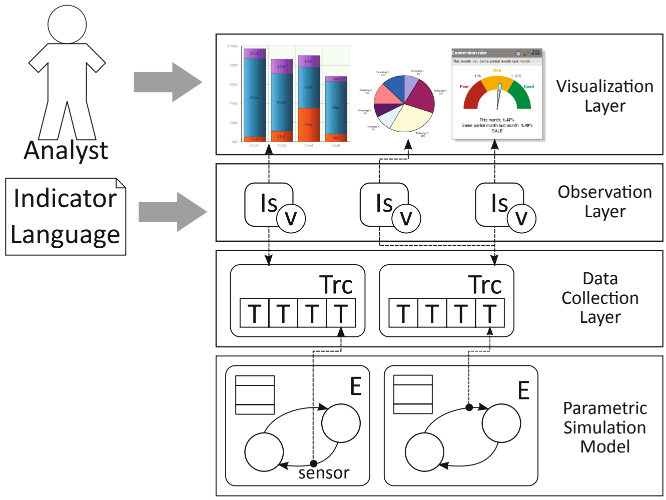


Fig. 5. Structure of observation model

```
1  indicator-category Performance;
2  specific-indicator StackedSearchNewClients {
3    categories -> Performance;
4    element ActivityInstance {
5        type -> "IAutomaticActivity";
6        filter -> "#self.process.parent.name == SearchNewClients";
7        observation-points {
8            timestamp initTime -> "execute";
9            timestamp finishTime -> "endExecution";
10       }
11   }
12   variables {
13       double[ ] automaticProcessingTime -> "ActivityInstance.finishTime -
14       ActivityInstance.initTime" groupBy "ActivityInstance.process.name";
15       string processName -> "ActivityInstance.process.name" groupBy
16       "ActivityInstance.process.name";
17   }
18   visualization {
19       title -> "Stacked search new clients";
20       bar-chart {
21           orientation -> horizontal;
22           stacked -> true;
23           bars {
24               bar "processName" -> {"automaticProcessingTime"};
25           }
26       }
27   }
28 }
```

Listing 1.1. Example of indicators definition

Listing 1.1 presents an example of the Indicator Language. Line 1 presents the definition of an indicator category. These categories classify the desired indicators so they can be grouped according to the Analysts interests (line 3). To specify an indicator, an Analyst first specifies the elements to monitor (line 4 - 11). In the sample case, only instances of automatic activities are monitored (line 5). For each element, we also specify a filter (line 6) to guarantee that sensors are only placed in the automatic activities of the SearchNewClient process and not on activities from other processes. Then, we identify the observation points for the element which corresponds to actions in transitions in the element's state machine (lines 8 and 9). After the elements to monitor, an analyst then specifies the variables and formulas to calculate the indicator (lines 12 - 17). Finally, variables are used to build the indicators representation. Lines 20 to 24 show that for the Search New Clients process indicator the results will be presented using a stacked bar chart.

4 Simulation Experiment

Experimentation in simulation is the ability to configure and test distinct scenarios to observe the outcomes of each simulation execution, and to relate these results to the models used as input. This serves to answer *what if* questions about the operation of the organization. We refer to an experiment as a particular configuration of a scenario (PSM and indicators), complemented with the definition of stimuli that represent influences to the model.

To configure PSMs, we currently use property files that set the values for certain attributes of the model. The aspects that can be configured using these property files range from selecting a specific probability distribution with given parameters to describe the amount of time an Operation takes to complete, or more complex behavior such as specifying a task assignment policy (taken from a set of predefined policies). Listing 1.2 presents an example of a configuration for the PSM introduced in the previous section. Line 1 specifies the distribution to calculate the response time for the operation Op_LoadPartnerClientsAsProspects and lines 2 and 3 present the parameters to calculate said distribution. The rest of the file configures similar properties for the other operations that support activities in the process.

```
1  Op_LoadPartnerClientsAsProspects.responseTime.distribution=Normal
2  Op_LoadPartnerClientsAsProspects.responseTime.mu=120000
3  Op_LoadPartnerClientsAsProspects.responseTime.sigma=20000
4
5  Op_CheckProspectByRegulation.responseTime.distribution=Triangular
6  Op_CheckProspectByRegulation.responseTime.min=10000
7  Op_CheckProspectByRegulation.responseTime.max=25000
8  Op_CheckProspectByRegulation.responseTime.mode=20000
9
10 Op_SegmentPerson.responseTime.distribution=Uniform
11 Op_SegmentPerson.responseTime.min=5000
12 Op_SegmentPerson.responseTime.max=8000
13
14 Op_SendMail.responseTime.distribution=Constant
15 Op_SendMail.responseTime.value=30000
```

Listing 1.2. Configuration example for elements of type Operation

On the other hand, a simulation experiment includes a *Stimuli* definition. Stimuli are influences over the model that affect some of the organization's elements at certain time intervals. Since an organization isn't an entity detached from its surroundings, people or events from its environment can affect its operation. For example, a possible stimulus for the BDLA scenario would be a client that initiates a process to open a product. Stimuli are defined using a *Stimuli Language*. Listing 1.3 shows a small example of stimuli for the PSM introduced in the previous section. The stimuli language allows defining who generates the stimulus (an element of the model typed *Stimulable* – e.g. a Person named `John`, line 2) and the action it triggers (e.g., `instantiateProcess`, line 4). Line 5 indicates that the action `instantiateProcess` requires specifying the name of the process to be started (`SearchNewClients`). Additionally, for each stimulus it is necessary to provide information about the time intervals it occurs. The first occurrence of the stimulus is defined by its initiation time, which can be at any specific time of the execution (e.g., `30.0 minutes` after the simulation began, line 9) or at the beginning (`0.0 milliseconds`). The following occurrences of the stimulus are inserted in the timeline according to a probabilistic distribution. Line 10 specifies that stimuli will be inserted using a Chi distribution with `5 degrees of freedom`. Finally, the termination condition indicates when to stop creating occurrences of the stimulus and can be specified as a certain time after the first stimulus is inserted in the timeline or as the maximum number of repetitions of the stimulus (e.g., `5 times`, line 16).

```
1  stimulus {
2      actor "John";
3      action {
4          name -> "instantiateProcess";
5          parameters {
6              string process -> "SearchNewClients";
7          }
8      }
9      time -> 30.0 minutes;
10     distribution {
11         chi {
12             degrees-freedom -> 5;
13             square -> false;
14         }
15     }
16     termination-condition {
17         repeat-until -> 5 times;
18     }
19 }
```

Listing 1.3. Example of stimuli definition

Once a simulation experiment has been completely defined, it can be loaded in the *Simulation Engine*. This engine runs the simulation while sensors gather information about the execution and store it in traces. After executing the simulation, traces are processed and indicators are calculated and presented using a *Visualization Tool*. The results can then be evaluated by *Business Analysts*, which propose new experiments until a proper architecture that satisfies the organization's needs is found.

5 EA Simulation Workflow and Validation

The elements presented in the previous sections are the building blocks of this paper's proposal. Figure 6 presents an ideal workflow to build and execute a simulation. First, Business Analysts define the simulation requirements, taking into account their interests in the project. From these requirements, *EA Metamodelers* design (or select from a repository) an EA SCM that matches them using a *Metamodeling Tool*. Said SCM must extend from the SMM to include simulation concepts.

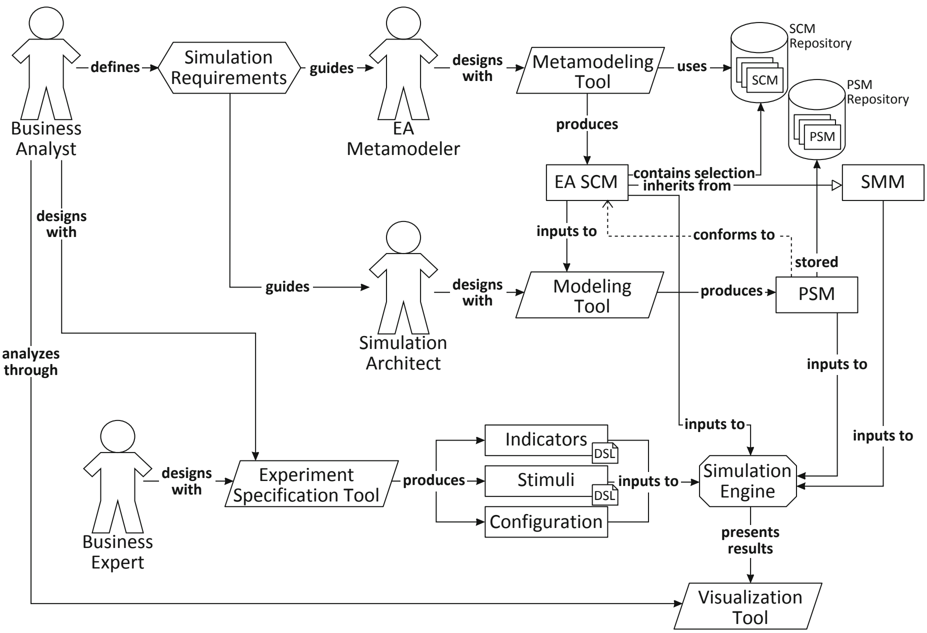


Fig. 6. A workflow for applying the proposed approach

After an SCM has been created, a *Simulation Architect*, guided by the requirements, builds a PSM using a *Modeling Tool*, which conforms to the EA SCM designed by the *EA Metamodeler*. Afterwards, Business Analysts use an *Experiment Specification Tool* to complete the definition of a simulation scenario by defining the desired indicators. With this same tool, *Business Experts* build experiments by specifying the model's Configuration and the Stimuli that will guide the simulation execution. Finally, all these artifacts are used as inputs for the *Simulation Engine*, which runs the experiments and presents the results through a *Visualization Tool* for Business Analysts to evaluate and make decisions accordingly.

As mentioned in section 3, we proposed a PSM using parts of a scenario called *Banco de los Alpes* (BDLA) [12]. BDLA is a banking initiative focused on the young adults market segment and on venture projects. It has approximately 25 processes distributed along 5 macroprocesses in its value chain. To validate our proposal, we use the following scenario/experiment, whose design cannot be explained due to a lack of space. We selected the SCM presented in figure 2 and a conformant PSM (figure 4), as well as a set of indicators (listing 1.1) that aim to studying the SearchNewClients process. For each run of the experiment, the engine created 3 instances of the process as specified in the stimuli definition (listing 1.3), which behaved according to the configuration presented in listing 1.2.

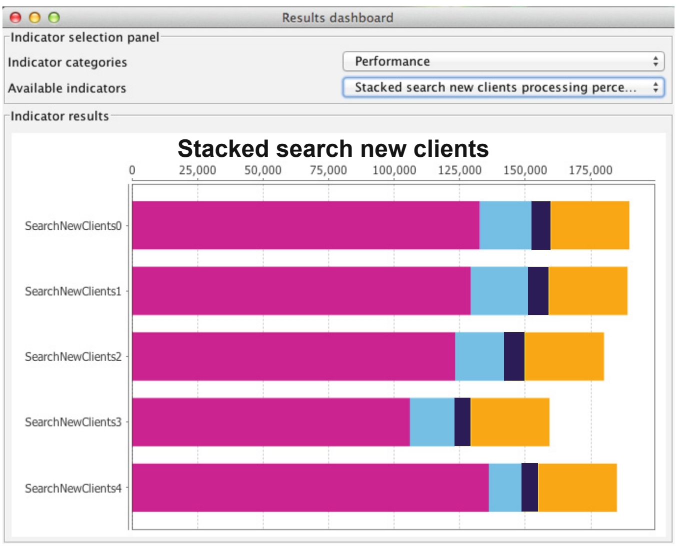


Fig. 7. Indicator of a simulation execution

The SearchNewClients process has solely automatic activities, but in our simulation they can take a significant amount of time to complete. Figure 7 presents the results of the defined experiment, where we can observe that the first activity the process executes (LoadProspectsFromPartners - Act1) corresponds to more than 65% of the completion time in every instance of the process. Table 1 shows the activities completion times in milliseconds. We can observe that the last activity of the process (NotifyApprovedProductsToProspects - Act4) has no effect over the differences in completion times of the instances since it always takes the same amount of time to complete (30,000 milliseconds). The average completion time of the process instances is 180,5756 milliseconds. We can observe that the process instances completion times ranged from 12% less than the average time up to 5% more. This happens because we used probabilistic distributions to calculate the response times of the process' activities, allowing the simulation to obtain different results in each experiment.

Table 1. Results of defined indicator in milliseconds

Process Instance	Act 1	Act 2	Act 3	Act 4	Total
SearchNewClients0	132,673	20,042	6,973	30,000	189,688
SearchNewClients1	129,314	22,081	7,596	30,000	188,991
SearchNewClients2	123,315	19,045	7,646	30,000	180,006
SearchNewClients3	105,987	17,371	5,883	30,000	159,241
SearchNewClients4	136,083	12,945	5,924	30,000	184,952

6 Related Work

EA analysis assesses certain criteria of an EA model [7]. By simulating an EA model, we can "enable enterprise architects to alter local strategies and observe their impact on the resulting global system behavior" [2]. Our proposal simulates an EA model in order to provide support for EA analysis by delivering information of the experiment execution to Business Analysts for their decision-making process. Taking into account the classification presented by Buckl et al in [3] for EA analysis, our proposal analyses *behavior statistics* and *dynamic behavior* for Body of Analysis dimension; *ex-post* and *ex-ante* for Time Reference; *indicator based* for Analysis Technique; and *non-functional* for Analysis Concern.

With this classification into consideration, we found various works with similar approaches. Among them, we found an approach based around system dynamics, agent-based and discrete events simulation [6]. They allow simulating elements from different domains or "views", which are of interest of distinct stakeholders. This is achieved by utilizing different simulation methods for each domain according to its simulation needs. Similarly, our approach supports the definition of metamodels for different domains, which can be composed into a single metamodel. But unlike their approach, our proposal uses only one simulation method in order to execute the EA model as a whole. Finally, similar to our approach, their proposal allows observing *behavior statistics*, *ex-post* and *ex-ante*, and *non-functional* analysis.

In [4], Crégut et al. animate models that conform to four fixed metamodels, where dynamism is expressed by defining execution states of the domain metamodel elements to control the progress of the overall animation. This approach is similar to out proposal, however, we represent dynamism using state machines that are external to the metamodel. This means that changes in the behavior of the elements, doesn't impact the metamodel and viceversa. Concerning simulation time management, [4] uses the states of elements to reflect time effects over the EA model, and time-passing information is recollected through events triggered by the elements of the model and expressed through event and traces metamodels, which are strongly connected to the EA metamodel and model. In contrast, our approach gathers information about the simulation using an observation model that is independent from the EA model. This represents an advantage since it leverages reusability of the model and metamodel. This approach focuses on both dynamic behavior and behavior statistics, as well as ex-post and ex-ante analysis, since the approach allows making decisions over the EA and adjusting the model to observe new behavior. Nevertheless, their approach doesn't specify how the animation results are presented in order to analyze them and make decisions.

Frank et al. [5] propose an indicator metamodel to evaluate non-functional properties of an organization within an Indicator System. Unlike our approach, they propose associating indicators to goals in order to assess if the system is supporting the organization's needs. Like our indicator language, they relate indicators to relevant elements of the organization, which are monitored to build indicators. On the other hand, Johnson et al. utilize Architecture Theory Diagrams (ATDs) [8] to assess the non-functional analysis dimension of the organization. We have studied this work and taken inspiration from it for our indicator language as well.

Finally, the main difference between our proposal and other EA simulation and analysis tools [4][5][6][8] is the flexibility for using ad hoc metamodels that include only domains of interest for the human experts, according to the project's needs. Other approaches define fixed metamodels that analysts are forced to use regardless of their own particular needs, which can introduce non-relevant information for the simulation into the models.

7 Conclusions

This paper proposes a model-based platform for building and simulating EA models to support decision making processes. The proposed platform offers the possibility of using EA metamodels, designed for particular projects, with any desired granularity, which is defined at the beginning of the project. This means that the metamodels only consider elements that pertain the EA analysis. The proposal also permits the definition of arbitrarily complex behaviors for each type of element involved, empowering the simulation. Additionally, allowing the definition of custom metamodels, we permit adapting the simulation to the particular definition of an organization, regardless of its business interests or requirements. This makes our proposal a suitable solution to make EA analysis for any enterprise.

Putting in place an EA simulation requires a number of different activities. Seeking to facilitate the definition of an EA simulation, we proposed an ideal workflow that aligns the simulation artifacts to the various activities of a simulation definition process. Furthermore, the characteristic of having multiple decoupled artifacts fosters reuse at several levels. One of those is the scenario/-experiment level: multiple experiments can be run on top of the same scenario, without requiring extensive modifications or re-designs. The results of all the experiments run over the same scenarios are presented using the same, comparable indicators. This facilitates the work of business analysts who participate in decision making processes and use the results of the simulation as inputs for analysis.

We have already considered several lines of future development for the platform. One of them is including complementing (and reusable) domain metamodels in the simulations. This should not be difficult to achieve on top of Cumbia since it has already shown its capacity to run multiple concern-specific workflow languages [10]. Additionally, although having custom-made metamodels can inccur in unnecessary extra effort, we plan to include means to transform existent EA metamodels and models designed with standard languages (e.g., BPMN or EPC), to facilitate the task of defining the simulation scenarios.

References

1. Banks, J., Carson, J.S., Nelson, B.L., Nicol, D.M.: Discrete Event System Simulation, 5th edn. Prentice Hall (2009)
2. Buckl, S., Matthes, F., Renz, W., Schweda, C.M., Sudeikat, J.: Towards simulation-supported enterprise architecture. In: Fachtagung Modellierung betrieblicher Informationssysteme (MobIS 2008), Saarbrucken, pp. 131–145 (2008)
3. Buckl, S., Matthes, F., Schweda, C.M.: Classifying Enterprise Architecture Analysis Approaches. In: Poler, R., van Sinderen, M., Sanchis, R. (eds.) IWEI 2009. LNBIP, vol. 38, pp. 66–79. Springer, Heidelberg (2009)
4. Crégut, X., Combemale, B., Pantel, M., Faudoux, R., Pavei, J.: Generative Technologies for Model Animation in the TopCased Platform. In: Kühne, T., Selic, B., Gervais, M.-P., Terrier, F. (eds.) ECMFA 2010. LNCS, vol. 6138, pp. 90–103. Springer, Heidelberg (2010)
5. Frank, U., Heise, D., Kattenstroth, H., Schauer, H.: Designing and utilising business indicator systems within enterprise models - outline of a method. In: Modelierung bietrieblicher Informationssysteme (MobIS 200) - Modellierung zwischen SOA und Compliance Management, Germany (2008)
6. Garret, C.: Understanding Enterprise Behavior using Hybrid Simulation of Enterprise Architecture. PhD thesis. Massachusetts Institute of Technology, USA (2009)
7. Johnson, P., Johansson, E., Sommestad, T., Ulberg, J.: A tool for enterprise architecture analysis. In: 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007), p. 142 (October 2007)
8. Johnson, P., Nordstrom, L., Lagerstrom, R.: Formalizing analysis of enterprise architecture. In: Interoperability for Enterprise Software and Applications Conference, p. 10 (April 2006)
9. Sánchez, M., Jiménez, C., Villalobos, J.: Model based testing for workflow enabled applications. Computación y Sistemas (2011)
10. Sánchez, M., Jiménez, C., Villalobos, J., Deridder, D.: Extensibility in Model-Based Business Process Engines. In: Oriol, M., Meyer, B. (eds.) TOOLS EUROPE 2009. LNBIP, vol. 33, pp. 157–174. Springer, Heidelberg (2009)
11. Sánchez, M., Villalobos, J., Romero, D.: A State Machine Based Coordination Model applied to Workflow Applications. Avances en Sistemas e Informática 6(1), 35–44 (2009)
12. TICSW Research Group Universidad de los Andes. Banco de los Alpes. Website (August 2011), http://sistemas.uniandes.edu.co/~losalpes/
13. TICSW Research Group Universidad de los Andes. EA Simulation Engine. Website (January 2012), http://cumbia.uniandes.edu.co/wikicumbia/doku.php?id=ea:start

Information Systems, Business and Law – Lessons Learnt

Anna Bobkowska

Gdańsk University of Technology Narutowicza, 11/12, 80-952 Gdańsk, Poland
annab@eti.pg.gda.pl

Abstract. This paper aims at exploring issues on the edge of information systems development, business process and decision making as well as related law and legal regulations. It describes several lessons learnt based on author's experience when applying a method of integrated legal and technical analysis in the development of an information system in the area of intellectual property protection. They contribute to the taxonomy of possible impacts of law on information systems as well as to the design of efficient methods which facilitate collaboration between analysts, lawyers and business decision makers.

1 Introduction

Information systems are deployed in almost every area of human endeavour. When they support business processes in industry or administration, they must conform to several business rules and procedures. Additionally, when the work of organizations or the areas of their activity are regulated by law, they must conform to relevant legal acts and regulations. Since the number of legal regulations increases and the legal regulations adhere often also to information systems, the support for the area on the edge of information systems, business and law becomes increasingly important.

Our research focuses on practitioner's perspective. From the perspectives of business analyst or software process engineer, there is a need for methods which allow to perform this interdisciplinary analysis effectively and efficiently. The methods should allow to overcome typical to interdisciplinary activities problems related to the fact that experts from different disciplines speak different professional language and that the same terms might have different meanings. Professionals have to collaborate in order to achieve common goals, thus coordination of actions performed by independent experts is another challenge. In our previous research, we have developed a method for integrated legal and technical analysis which was based on detailed analysis of requirements resulting from law in both lawyer and software engineer perspectives [2]. We have also performed case studies in which we have shown concrete examples of the impact of law on several kinds of requirements to information systems [3].

The goal of this paper is to present lessons learnt when applying the method of integrated legal and technical analysis in real research project. The system for plagiarism detection was chosen because intellectual property rights area has a great

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 470–477, 2012.

potential for legal analysis. The lessons learnt contribute to better understanding of the phenomena on the edge of information systems, business and law. With this knowledge, one can design more mature methods, formalisms and tools.

Related work is discussed at Requirements Engineering and Law (RELAW) workshops [1] which attempt to identify challenges [6], to design representations of legal regulations [4] and to increase collaboration between lawyers and engineers by the means of standardizing terminology between law, engineering and business.

The paper is structured as follows. Section 2 presents methodological remarks and case description. Sections 3-8 present lessons learnt together with observations on the basis of which they were formulated and discussion of their novelty and limitations in generalization. Section 9 draws conclusions.

2 Methodological Remarks and Case Description

The method of study is a case study of application of integrated legal and technical analysis in the real complex research project. The lessons learnt were collected on the basis of observation, reflection and reaction to anomalies. This section describes shortly the following:

- The method of integrated legal and technical analysis;
- The case which was studied;
- Remarks on collecting and describing lessons learnt.

According to the method of integrated legal and technical analysis, on the basis of a vision of the system, legal analysis is performed by a lawyer in parallel with technical (business and functional) analysis performed by an analyst. In the next step, the transformation of legal regulations to systems requirements is made in collaboration between the lawyer and the analyst. Specification of requirements is the duty of the analyst. Finally, the requirements specifications is validated from business, technical and legal perspectives.

The case of the system is Intellectual Property Protection System developed at the university as a research project. The goal of the system is to support detection of plagiarism on the basis of similarity report produced after running advanced and innovative algorithms for document similarity detection. There are no constrains on documents to be checked for plagiarism. They can be Master Thesis, assessment works by students, or research papers to be published in conference proceedings or in journals. There are no constraints on the users, so one can assume anyone will have access to the system: students, university employees, deans, editors of conference proceedings or journals and authors.

The lessons learnt were collected when performing integrated legal and technical analysis in the case of Intellectual Property Protection System. Reflections on the application consistent with presumptions as well as reflections on anomalies and reactions to them were described. The descriptions of lessons learnt are structured as follows: a statement of a lesson learnt, observations which were the background for formulating the lesson, and discussion of novelty of findings, confirmation to known facts, limitations in generalization as well as comparison to knowledge gained in our previous case studies.

3 Multiple Kinds of Relevant Legal Acts and Regulations

3.1 Lesson Learnt

Need to consider fragments from multiple legal acts and regulations with several kinds of relationships between information systems, business and law.

3.2 Observations

In this case, the lawyer has identified five relevant legal acts and three related executive regulations. Two of the acts with related executive regulation adhere to organization of higher education systems as well as the rules and procedures of granting degrees at universities. Thus, they can be classified as regulation of the organization in the domain. The next act is related to intellectual property rights to pieces of work. Master Thesis, books and papers are pieces of work according to this act. This type of impact can be called regulation of area of data under processing. Other legal acts and regulations were related to Privacy Policy and Database Protection. They apply to all information systems which process personal data.

3.3 Discussion

This case shows that the impact of law on the system via the domain can be differentiated e.g. organization of domain, business processes in organization, or area of data under processing. This finding contributes to the taxonomy of possible impacts of law on business and information systems.

The limitation in generalization is adherence to Polish law only. Although higher education and intellectual property rights are universal areas, the number and type of regulations in each country can be different. Especially, when acting in Anglo-Saxon tradition of case law, one might rather collect precedents than analyze legal acts. However, the fact of having several sources and possible kinds of legal constraints generalizes also to other systems in this domain and systems in other domains.

4 Deployment Dependent on Legal Context of Application

4.1 Lesson Learnt

Relevance of legal acts strongly depends on the context of application of information systems - even when scope of required functionality in several application areas seems the same, the context of application might set legal constraints on deployment.

4.2 Observations

The legal acts and regulations listed in section 3, were identified for application of the information system for the use at the university in computer science area. According to Polish law, patents are not granted in this area. Patent agents claim that when applying it in different branch of engineering, the appropriate legal regulations related to patent systems should be analyzed as well. Additionally, when using this system in research commercialization area, the law adhering to industrial property and unfair competition should be taken into account.

Similar system could be used in a publisher's house, where the legal acts related to higher education are not in force, but legal regulations related to publishing are in force instead. Finally, there is a question about commercial vs. non-commercial use of the system.

4.3 Discussion

It's a strange phenomena from software engineer's point of view that the system which implements advanced algorithms and which has passed all tests might have limits in deployment resulting from legal constraints.

When attempting to answer the question of whether it generalizes to other projects, it seems that it depends on the type of system. In case of a system for the single application area, the problem will not appear. However, when a system might be applied in several diversified areas, there is a need for management of the impact of the legal context on the scope of allowed application of the system.

5 Extensive Impact of Law on Several Kinds of Artifacts

5.1 Lesson Learnt

Law has impact on several artifacts of business, information system and its development process. It can be more extensive than expected.

5.2 Observations

This part summarizes the impact of law on business, information system and process. In this case, legal analysis had impact on the following artifacts:

- Vision - Legal analysis results suggest that a general system for plagiarism detection has a high risk of conflicting with abiding law; Situation is very different when one treats the system as a non-commercial tool available only to authorities of the university responsible for academic degrees which allows them to fulfill their legal obligation (see also section 4);
- Functionality - The most interesting example of the impact of law on functionality of the system is the requirement to collect, store and check

permissions of authors of copyright owners for using their piece of work on several fields of operation;

- Business process - There is a need to define internal business processes leading to statement of plagiarism suspicion which are consistent with legal rules of conduct in case of plagiarism suspicion and proper evaluation of the results of similarity detection by the information system (see also section 6);
- Software development and exploitation processes - Laws adhering to Database Protection and Privacy Policy with related executive regulations set up constraints on exploitation, e.g. formal requirements to administrators or the way of database administration; as well as software development process, e.g. requirements on documentation, necessity to register databases which contain personal data to General Privacy Inspector during the deployment of the system.

5.3 Discussion

The extensive scope of impact of law on the artifacts confirms the importance of legal analysis during the development of information systems.

The limitations in generalization are related to analysis of this single case in highly legalistic area. Comparing to our previous case studies, no explicit impact of law on scope of functionality, data stored in the system, algorithms of processing data, quality requirements or other legal requirements has been discovered. On the other hand, the impact on vision, functionality and business processes, is more significant than minor constraints or corrections.

6 Misunderstandings Resulting from Different Backgrounds

6.1 Lesson Learnt

Misunderstandings between lawyers and software developers are caused by discrepancy in viewing phenomena or using the same terms with different connotations by experts with different background.

6.2 Observations

Legal analysis of information systems is an interdisciplinary phenomenon and several misunderstandings and discrepancies may result from different views of the professionals. In this case, we have observed this effect in the following dimensions:

- parties and roles performing their duties vs. users manipulating objects,
- a document vs. a piece of work, and
- legal rules and procedures vs. internal business rules and procedures.

Software engineers focus mainly on the construction of systems and providing functionality for users while many administrative legal regulations describe organizations in terms of parties and roles as well as related duties and procedures of document circulation.

There is a difference between a document (object in a system, item for processing) and a piece of work (*pl. utwór*, in legal sense item for intellectual property right protection). Independently of the type of document (Master Thesis, book, paper etc.) or copyright owner and related permissions, documents are stored in the system in the same way and they are effectively processed with the same algorithms. From the legal perspective, they constitute several cases which must be considered separately and sometimes might disable processing or even deployment of the system.

Both legal acts and university regulations use terms of rules and procedures. However, they might be not exactly the same rules and procedures. Sometimes law defines a kind of framework for internal rules and procedures in organization. In this case, legal regulations describe rules of conduct in case of plagiarism suspicion such as explanatory proceeding, possible consequences of terminating employment with university staff, or statement of invalidity of conduct leading to granting a degree. They can be applied when suspicion of plagiarism has been already stated. On the other hand, a similarity detection report itself cannot be proof of plagiarism. There is a need to define roles responsible for evaluating the similarity detection report from the semantic perspective and to define for them the rules of conduct.

6.3 Discussion

In our opinion, these are only examples of possible implicit problems caused by the use of similar terms by professionals in law, business and information systems. There is a large potential in exploring them in details in order to achieve maturity in dealing with the issues on the edge of them. The roots of the problem are in missing the fundamentals of other disciplines by the experts. Thus, the solution is in training interdisciplinary experts who are sensitive to specifics of all the involved areas.

7 Comments on Methods of Interdisciplinary Analysis

7.1 Lesson Learnt

Methods of integrated legal, business, and information system analysis facilitate the work and decrease complexity when they fit to the circumstances of application. As these circumstances can be diversified, the best solution can be provided by several methodological patterns with the guidelines on their application.

7.2 Observations

It has appeared that the lawyer met difficulties with performing entire legal analysis. Thus, the phase of legal analysis was divided into two sub-phases: identification of legal issues made by the analyst and lawyer's explanations. The analyst has prepared thirty nine questions and the lawyer prepared thirteen pages of explanations with comments on relationship between answers and related legal acts and regulations [5]. The number of pages of the relevant legal acts and regulations is two hundreds ninety one. The method has really facilitated the work and the resulting report was a good artifact for communicating the results and following discussions.

7.3 Discussion

The primary version of the method can be recommended to collaboration with lawyers having the knowledge of fundamentals of software engineering as well as some experience in working on the edge of law and information systems. When the lawyers have little experience in working in this area, a second variant of the method (with questions to lawyers) is more appropriate.

The method was useful and it is likely that it will be useful also in similar cases. The first limitation in generalization is the fact that it applied to a single project in contrast to software product lines which must meet other challenges and they require a more advanced infrastructure. The second limitation is the fact that system was developed for an organization with running business processes. When starting a new business a more advanced and more flexible analysis would be required.

8 Need to Change Approach to Multi-variant Analysis?

8.1 Lesson Learnt

In complicated cases, multi-variant analysis made in interaction between decision makers, business analysts, lawyers and information system experts is recommended instead of legal analysis of the impact of law on information systems

8.2 Observations

Decision makers have shown discontent when hearing about suggested changes in vision. They claimed that lawyers exaggerate and that such law is against the progress. They have suggested that lawyers should search for possible solutions instead of discovering constraints.

8.3 Discussion

The root of this problem is in the misfit between law which was established before information technology era and new possibilities of innovative application of this technology. Some regulations although formally in force may belong to, so called, dead law. Secondary problem is in differences of positions of decision makers tending to achieve an innovative product and lawyers responsible for conformance with law in the project.

It can be resolved on two ways. First, legislators adjust the law for the use of information technology. In fact, it already takes place in several legislative areas, e.g. regulations related to accounting, banking, administration. Second, lawyers involved in software projects may search for interpretation to legislators or branch committees. The consequence for the project is the replacement of legal analysis with multi-variant interdisciplinary analysis similar to feasibility studies in project management practice.

9 Conclusions

Real project cases usually exceed neat theories and methods. They provide a more diversified view on the phenomenon under exploration. On the other hand, theories and methods structure the goals and actions, enable efficient performance and provide framework for improvement. So it was in this case. The applied method facilitated work of performing the analysis. However, in action one could see also its limitations and additional phenomena which we have described as lessons learnt. They are related to the multiple kinds of impact of the relevant legal regulations on information systems, the dependence of the deployment on the legal context of applications, the extensive impact of law on several kinds of system artifacts, the sources of misunderstandings between analysts and lawyers resulting from different viewing of the same phenomena, the methods of interdisciplinary analysis and considerations about the right approach depending on circumstances of application.

These lessons learnt contribute both conceptually and methodologically to the base of knowledge about the phenomena on the edge of information systems, business and law. The understanding of fundamentals increases awareness of issues when performing activities related to legal analysis and the methods guide the process of legal analysis. Both of them support practitioners in their actions. Together with other original lessons learnt, they can facilitate development of the taxonomies of related issues and the design of mature methods which allow for effective and efficient conduct of this interdisciplinary analysis.

Acknowledgements. The work was partially supported by MAYDAY EURO 2012 project, Operational Program Innovative Economy 2007-2013, Priority 2 “Infrastructure area R&D”.

References

[1] Anton, A.I., Breaux, T.D., Karagiannis, D., Mylopoulos, J.: First International Workshop on Requirements Engineering and Law (RELAW). In: Proceedings of the Workshop on Requirements Engineering and Law, RELAW 2008 (2008)
[2] Bobkowska, A., Kowalska, M.: On efficient collaboration between lawyers and software engineers when transforming legal regulations to law-related requirements, vol. 18. Annals of Faculty of ETI of Gdańsk University of Technology (2010)
[3] Bobkowska, A., Kowalska, M.: Wpływ prawa na systemy informatyczne - studia przypadków. In: Zeszyty Naukowe Wydziału Elektroniki, Telekomunikacji i Informatyki Politechniki Gdańskiej, vol. 19 (2010)
[4] Breaux, T.D., Vail, M.W., Antón, A.I.: Towards Regulatory Compliance: Extracting Rights and Obligations to Align Requirements with Regulations. In: Proceedings of 14th IEEE International Requirements Engineering Conference, RE 2006 (2006)
[5] Kowalska, M.: Lawyer’s explanations. Technical Report on Validation of Legal Analysis Method in Case of Legal Analysis of Intellectual Property Protection System (2011)
[6] Jörg Leuser, J., Penzenstadler, B.: Complying with Law for RE in the Automotive Domain. In: Proceedings of the Workshop on Requirements Engineering and Law, RELAW 2008 (2008)

Towards Formal Expression of Business Rules Written in Polish

Bogumiła Hnatkowska and Anna Błasiak

Wroclaw University of Technology, Institute of Informatics, Wyb. Wyspianskiego 27, 50-370 Wrocław, Poland
Bogumila.Hnatkowska@pwr.wroc.pl, 164749@student.pwr.wroc.pl

Abstract. Business rules are very important part of a business model. They should be defined in an unambiguous, understandable way, especially when they are going to be implemented in a software system. There are some attempts addressing that issue but none of them considers business rules written in Polish. The paper presents existing methods and tools that support business rules written in English, and the roadmap to adapt them to Polish language.

Keywords: Business rules, business rules formalisms, MDA, CIM, SBVR, Polish.

1 Introduction

Business rules are very important part of a business model, and next important part of requirements specification for software system. However, there is a big gap between these two worlds, i.e. a specific domain which is known to domain experts, and the IT domain in which business knowledge first must be correctly understood, interpreted, and represented in the manner that fits the needs of software development.

The language in which business rules are expressed must be proper for their audience. At the business level business rules are expressed in natural language, typically in declarative way [1,2]. Natural language brings obvious benefits – it is understandable for all interested parties – but also commonly known problems: misinterpretation, incomprehension etc.

In the real enterprise there are plenty of business rules defined. In such a case completeness and consistency checking of business rules specification is challenging task. The natural solution of the above mentioned problem is to apply tools that will not only help in gathering and checking business rules but also offer support to software development. Model Driven Architecture (MDA) is an approach to software development which seems to be ideal for that [4].

The aim of the paper is to present the current state of art about tools and methods that are suitable to: (a) represent business rules in natural language, and (b) translate natural language specification into formalisms that are suitable for further development. This is the first stage of the research which is going to propose such solution for Polish language. The existing solutions support mainly English which is far less flexible, and has less complex grammar than Polish. According to [31] "the Polish

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 478–485, 2012.

language is one of the most difficult languages to learn with its tongue-bending pronunciation, complex gender system, seven cases, ‘aspect’ as a grammatical category of the verb and a tendency to avoid internationalisms for ‘real’ Polish words”.

The paper is organized as follows. Section 2. provides the definition of basic notions. Section 3. explains the role of business rules and shortly presents languages used for business rules expression in the context of software development with MDA approach. Next, Section 4. concentrates only on methods allowing to express business rules in natural languages and presents solutions and standards in this area. Section 5. gives an overview of tools supporting business rules processing at different levels. In this section also some useful tools for Polish are presented. Last Section 6 presents the roadmap of future work.

2 Business Rules Overview

Business model defines basic notions from a given domain, the relationships between the notions and the way in which they are constrained. A business rule is a statement that defines or constrains some aspect of the business. It is intended to assert business structure or to control or influence the behavior of the business [5]. According to [1,2] business rules should be expressed in *natural language*, in *declarative way*.

A business rules can belong to a specific type, however, there are no commonly accepted, one business rules classification. But most of works (e.g. [1–3, 5]) introduce following categories of business rules:

- *terms* – a definition of something,
- *facts* – a relationship between terms,
- *constraint rules* – specify policies or conditions that restrict object structure and behavior; they may always apply (invariants) or apply only under certain conditions:
  - *operation constraint rules* – typically expressed as pre and/or post conditions,
  - *structure constraint rules* – specify polices about attributes’ domain, class instance numbers etc.
- *derivation rules* – specify policies or conditions for inferring or computing facts from other facts:
  - *inference rules* – specify that if certain facts are true, a conclusion can be inferred,
  - *computation rules* – derive their results by way of processing algorithms, a more sophisticated variant of inference rules,
- *stimulus and response rules* – any rule that tends to take some action automatically when a relevant event occurs. Such action might be to create or delete data, enable or disable another rule, set some value, execute some program or procedure.

3 Business Rules in the Context of MDA

Model Driven Architecture (MDA) is a standard proposed by OMG presenting a rigorous approach to software development in which models play a crucial role [4].

MDA introduces three types of models, presenting the system on different abstraction levels. Each subsequent model is a refined or extended version of the previous one. The models are as follows (see fig. 1):

- *Computation Independent Model* (CIM), which takes care of an environment in which the system will operate. This model mainly presents business model and it gathers business rules. It also includes requirement specification. There are some attempts to represent business rules in natural language at that level, e.g. RuleSpeak [6], Attempo Controlled English (ACE) [7,8], SBVR Structured English (SBVRSE) [9]. None of them is defined for Polish.
- *Platform Independent Model* (PIM), which presents the solution in the form that hides all platform specific details, and is so general that could be applied at any targeted platform. At that level business rules are expressed with the use of formal notations [30] (e.g. RuleML, Rule Interchange Format), and typically are spread among many UML diagrams [10]. Object Constraint Language (OCL) [11] is also used for their representation.
- *Platform Specific Model* (PSM), which presents the solution in terms specific to a selected platform. At that level business rules are typically implemented in the code, sometimes as a separate layer (e.g. [12]) or with the use of design patterns (e.g. [13]). Realization of business rules could be also delegated to special engines, e.g. JRules [14] or database servers, e.g. [15].

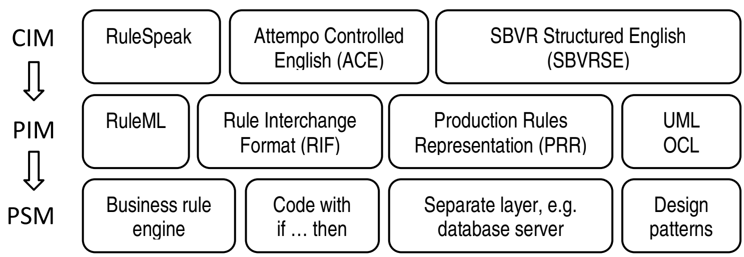


Fig. 1. Business rules' representation in MDA approach

The paper concentrates on business rules expressed at CIM level. The approaches used for that – pointed out at the fig. 1– are described in greater detail in subsequent section.

In the MDA transformations between models play important role. They are typically defined in terms of meta-models that provide abstract syntax for model representation. To enable CIM business rules to PIM business rules transformation the proper meta-models for them need to be either selected or elaborated from the beginning. For example, MOF is the meta-model for UML language, and Semantics of Business Vocabulary and Business Rules (SBVR) is the meta-model for SBVRSE.

4 Attempts to Express Business Rules in Natural Language

There are some attempts to expressing business rules in natural language, e.g. [6,7,9]. Most of them consider English as a native language of a domain expert. The attempts support the expert in different ways. Sometimes they take a form of a guideline that only help to write rules in more readable and clear manner. RuleSpeak belongs to that group [6]. Other attempts restrict the expert's language by defining the exact structure (grammar) of business rules. ACE [7,8] and SBVRSE [9] are the representatives of that group.

4.1 RuleSpeak

RuleSpeak introduces a concept of a sentence form which is defined as "a basic pattern or template in natural language that can be used to express a certain kind of business rule in a consistent, well-organized manner" [6]. It distinguishes two kinds of business rules (called here guided statements): business rule statements, and statement of advices. Business rule statements enable to express that something is required, disallowed or conditionally allowed. Statement of advice permits to express that something is allowed or not required.

The sentence form should be constructed in a specified manner with the use of so called rule keywords. For example, every business rule statement must use one of the following words: "must" or "only", while a statement of advice must use: "may" or "need not". Statements of advice can't be directly mapped to the taxonomy of business rules presented in Section 2.

At that moment RuleSpeak does not introduce the concept of glossary for the definition of terms used in guidance statements.

4.2 ACE

Attempto Controlled English (ACE) is a controlled language elaborated at University of Zurich [7,8]. ACE enables to express all types of business rules, starting from constrains to stimulus/response rules.

ACE is a formal language in the sense, that texts written in it "are computer-processable and can be unambiguously translated into discourse representation structures (DRS), a syntactic variant of first logic" [8]. DRS representation can be further translated into other languages, and RuleML is among them.

The grammar of ACE defines the syntax of ACE texts. ACE's syntax is expressed as a set of construction rules. The meaning of syntactically correct sentences is described as a set of interpretation rules. The grammar rules are build on a vocabulary which consists of predefined function words (etc. articles, conjunctions), predefined fixed phrases (e.g. 'there is') and user-defined content words.

Unfortunately, the vocabulary, and grammar rules are defined only for English, and it is doubtfully they can be easy adopted for another language, especially as morphologically reach as Polish. However, some ideas about acceptable sentence structures, and potential problems with their interpretation can be get from ACE experiences.

There are also some useful tools available for ACE. Attempto Parsing Engine (APE) is one of the most important. It checks the correctness of ACE texts, and – when they are correct – is able to generate syntax tree or DRS.

4.3 SBVRSE and SBVR

Semantics of Business Vocabulary and Business Rules (SBVR) is an OMG standard, providing a meta-model for documenting the semantics of business rules and vocabulary [9]. Its main goal is to enable defining business elements in friendly manner, preferably in natural language. It facilitates automation of business rules in software systems by structuring them as logical semantics formulation. SBVR standard also defines CMOF meta-model and XMI schema for interchange of business vocabulary and rules between software tools.

The standard document distinguishes two kind of business rules: structural and operative ones. Structural business rules "are rules about how the business chooses to organize (i.e., 'structure') the things it deals with" [9]. In turn, operative business rules "are rules that govern the conduct of business activity. In contrast to structural rules, operative rules are ones that can be directly violated by people involved in the affairs of the business" [9]. SBVR specifies also other elements of guidance such us advices or business policy.

Business rules in SBVR are always build on fact types, and the fact types are build on concepts expressed by terms. Fact type is an equivalent to the fact and concept is an equivalent to term in taxonomy presented in Section 2.

The major downsides of this meta-model are: its significant complexity [30], and the lack of support for standard arithmetic beyond numeric comparison [32].

It should be emphasized that SBVR distinguishes the *meaning* of rules and vocabulary from their representation (*expression*) thereby enabling representation of the same rule, term or fact in different languages. The standard itself is described with the use of not-normative English notation called SBVR Structured English (SBVRSE), which is a limited subset of English language. SBVRSE is only one of possible notations for SBVR representation. Another potential notation could be SVBR Structured Polish.

5 Business Rules Support in the Context of MDA

The planned final result of the work is to elaborate formal representation of business rules written in (controlled) Polish. This is the first step to enable models verification, and CIM-PIM (or PIM-CIM) model transformations. The fig. 2 demonstrates the scope of interests (marked in gray). It is assumed that transformations and verifications are done automatically. Because SBVR is native language independent standard it is considered as the first choice option for formal representation of business rules.

There are few editors of SBVR rules, e.g. SBeaVeR [17], VeTIS [18], RuleXpress [19], KnowEnterprise [20]. The first one allows to create, validate and verify the business rules and vocabulary written in SBVRSE, but it does not generate any formal representation or PIM model. Similar functionality is offered by two commercial tools: KnowEnterprise and RuleXpress, which additionally translate business vocabulary into SBVR standard. Noteworthy is another open source tool – VeTIS, which not only covers SBeaVeR functionality but also enables a user to generate UML class diagrams from business rules and vocabulary.

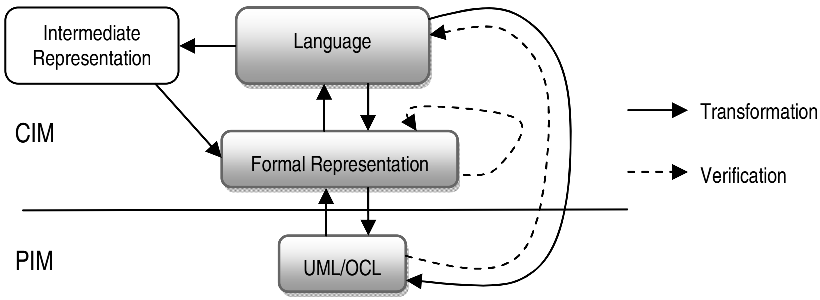


Fig. 2. Existing and planned areas of tool support in the context of business rules

SBVR standard was also used in NL2OCLviaSBVR tool [21], which uses it as an intermediate step in transformation from business rules written in English to OCL constraints. The tool does not provide any editor and assumes that vocabulary terms and facts are defined not in controlled English but in UML class diagram.

There are also some tools doing PIM-CIM transformations, e.g. UML2SBVR [22]. The tool takes an XMI file (representing UML class diagram) and generates another XMI consistent with SBVR standard.

Finally, Eclipse community has set up a SBVR/MDT project [23] to provide SBVR meta-model implementation and sample tools to support developers working with SBVR standard. Currently they managed to create ecore file based on CMOF meta-model and generate classes from it to enable loading and saving XMI files compliant with SBVR exchange document format.

In the context of the subject of this work (formal expression of business rules written in Polish), the problem of texts parsing and translating them into SBVR instances is the most challenging task. The problem is solved in different manners for SBVR Structured English. For example, the approach presented in [24] uses AI methods to parse rules written in SBVRSE and represent them as a model conforming to Syntax meta-model. This model is then transformed to model instantiating SBVR meta-model with usage of ATL tool. In the second approach [25] authors try to represent business rules in the form similar to semantic formulations exemplified in SBVR specification. The output of their parser can be easily transformed to XMI format. Finally, VeTIS developers have directly implemented some parsing rules in order to recognize SBVR elements in files containing vocabulary and business rules and then to represent them as objects – instances of SBVR classes. The second and third approach use deterministic algorithms that could cause some mistakes while parsing SBVRSE, because – as authors of the first approach pointed out in [24] – this notation makes it possible to interpret one sentence in different ways. That is why they search for a solution using advanced AI techniques.

There are also some tools supporting processing of Polish language. Morpheus [26] is one of the most useful offering morphological analysis of Polish sentences. For a given word it provides: all basic forms of the word, part of speech, grammar category values assigned to part of speech (e.g. gender, case etc.). It uses grammatical vocabulary of Polish language [27], which contains complete grammar characteristics of about 245 000 Polish words with their inflection forms (over 4 millions).

Another useful tool is TaKIPI [28] – a Polish language tagger, which determines for a word in some text its morpho-syntactical description and interprets its meaning depending on context. TaKIPI is also integrated with another tool Guesser [33], which performs morpho-syntactical analysis on unfamiliar words.

At the moment there is also a Polish version of WordNet – plWordNet (pl. Słowosieć) [29].

6 Summary

The paper deals with the problem of formal representation of business rules written in natural language. The state of art of existing approaches and tools was broadly presented. At that moment there are no solutions that give support for Polish language.

As a part of further work it is planned to adapt SBVR for Polish language, and to elaborate SBVR Structured Polish (SBVRSP) – the equivalent of SBVRSE – together with supporting tools. Because SBVR standard introduces many types of element of guidance, first structural business rules will be taken into consideration. Realization of that task requires elaboration of grammar and style rules for selected subset of Polish language (RuleSpeak for Polish). The grammar and style rules should be verified whether their application provides unambiguous, and understandable business rule definitions. Having all above mentioned steps done it will be possible to develop SBVRSP editor. The next and most complicated step is a parser development. The parser should translate business rules expressed in Polish into SBVR and maybe other formal notations.

SBVR standard was selected because it allows to separate business rules representation (Polish language) from their meaning. As a result existing tools, useful in the MDA approach, that take SBVR compliant input can be used without further modifications.

References

1. Ross, R.G.: Principles of the Business Rule Approach. Addison Wesley (2003)
2. Morgan, T.: Business Rules and Information Systems – Aligning IT with Business Goals. Adisson-Wesley (2002)
3. von Halle, B.: Business Rules Applied – Building Better Systems Using the Business Rules Approach. Wiley (2002)
4. MDA Guide v. 1.0.1., `http://www.omg.org/mda`
5. Business Rules Group: Defining Business Rules – What are they really?, `http://www.businessrulesgroup.org/first_paper/br01c1.html`
6. Ross, R.G.: RuleSpeak® Sentence Forms: Specifying Natural-Language Business Rules in English. Business Rules Journal 10(4) (2009)
7. Attempto Project, `http://attempto.ifi.uzh.ch/site/`

8. Fuchs, N.E., Kaljurand, K., Kuhn, T.: Attempto Controlled English for Knowledge Representation (2008), `http://attempto.ifi.uzh.ch/site/`
9. Semantics of Business vocabulary and Rules (SBVR), OMG Standard, v. 1.0, (2008), `http://www.omg.org/spec/SBVR/1.0/`
10. OMG Unified Modeling Language (OMG UML), Superstructure Version 2.3, (2010), `http://www.omg.org/spec/`
11. OCL 2.0 OMG Final Adopted Specification, `http://www.omg.org/docs/ptc/05-06-06.pdf`
12. Hnatkowska, B., Kasprzyk, K.: Business rules modularization with AOP. Przegląd Elektrotechniczny R 86(9), 234–238 (2010)
13. Arsanjani, A.: Rule Object 2001: A Pattern Language for Adaptive and Scalable Business Rule Design and Construction. In: 39th International Conference and Exhibition on Technology of Object-Oriented Languages and Systems (2001)
14. WebSphere ILOG JRules RBMS, `http://www-01.ibm.com/software/integration/business-rule-management/jrules-family/`
15. Koppelaars, T.: RuleGen Quickstart Guide. RuleGen BV (2009)
16. KDM Analytics, Semantics of Business Vocabulary and Business Rules (SBVR), `http://www.kdmanalytics.com/sbvr/sbvr_intro_3.html`
17. `http://sbeaver.sourceforge.net/`
18. Kaunas University of Technology Department of Information Systems, Creating UML&OCL Models from SBVR Business Vocabularies and Business Rules. VeTIS User Guide. Kaunas (2009), `http://www.magicdraw.com/files/manuals/VeTISUserGuide.pdf`
19. RuleArts, LLC, RuleXpress: The business tool for expressing and communicating business rule, `http://www.rulexpress.com/`
20. KnowGravity Inc.: Comprehensive support for the business perspective of Model Driven Enterprise Engineering, `http://www.knowgravity.ch/eng/pdf-e/KnowEnterprise-BU%20E.pdf`
21. Bajwa, I.S., Lee, M.G., Bordbar, B.: SBVR Business Rules Generation from Natural Language Specification. In: AAAI Spring Symposium 2011 – Artificial Intelligence 4 Business Agility. AAAI, San Francisco (2011)
22. Transforming UML Conceptual Schemas to SBVR: Tool description, `http://jordicabot.com/research/SBVR/index.html`
23. SBVR/MDT, `http://wiki.eclipse.org/MDT-SBVR`
24. Kleiner, M., Albert, P., Bézivin, J.: Parsing SBVR-Based Controlled Languages. In: Schürr, A., Selic, B. (eds.) MODELS 2009. LNCS, vol. 5795, pp. 122–136. Springer, Heidelberg (2009)
25. Marinos, A., Moschoyiannis, S., Krause, P.: An SBVR to SQL Compiler. RuleML Challenge (2010)
26. Morpheus morphological analyzer, `http://sgjp.pl/morfeusz/morfeusz.html`
27. Saloni, Z., Grzuszczyński, W., Woliński, M., Wołosz, R.: Słownik gramatyczny języka polskiego. Wydanie I, Wiedza Powszechna, Warszawa (2007), `http://sgjp.pl/`
28. TaKIPI Tagger for Polish language, `http://nlp.pwr.wroc.pl/takipi/`
29. PlWordNet version 1.6, `http://plwordnet.pwr.wroc.pl/wordnet/`
30. Linehan, M.H.: SBVR Use Cases. Information Systems Journal, 182–196 (2008)
31. `http://www.bbc.co.uk/languages/polish/soap/background.shtml`
32. Linehan, M.H.: Ontologies and Rules in Business Models Rules in SBVR Vocabularies in SBVR. Constraints, 149–156 (2008)
33. Piasecki, M., Radziszewski, A.: Polish Morphological Guesser Based on a Statistical *A Tergo* Index. In: Proceedings of the International Multiconference on Computer Science and Information Technology, AAIA 2007, pp. 247–256 (2007)

Modelling and Verification of Interorganizational Workflows with Security Constraints: A Petri Nets-Based Approach

Oana Otilia Captarencu

"Al. I. Cuza" University, Faculty of Computer Science
Gen. Berthlot St, No 16, 740083 Iasi, Romania
otto@infoiasi.ro

Abstract. Interorganizational worfklows represent workflows which involve several business processes, belonging to different organizations, which need to coordinate thier actions in order to achieve a common goal. This paper proposes a Petri net model which permits the modelling and verification of interorganizational workflows. The model allows the explicit representation of the organizational dimension of each component workflow, the shared use of resources among different organizations as well as the specification of security constraints. A property of soundness, which describes the correct execution of the interorganizational workflow, is defined and proven decidable.

Keywords: interorganizational workflows, modelling, verification, security constraints, Petri nets.

1 Introduction

A workflow is defined as a set of coordinated tasks that take place inside an organization in order to achieve a common business objective. Tasks may be carried out by resources (such as human users) and they involve specific data. Thus, several aspects (dimensions) of a workflow can be identified: the process dimension (which specifies the order of tasks), the organizational dimension, the data-flow dimension. Interorganizational workflows consist of several workflows belonging to different organizations, which execute independently but need to coordinate their actions at certain points, in order to accomplish a common goal.

One direction in the current research related to interorganizational workflows aims at finding suitable formal models to permit their specification and the analysis of properties such as the correct execution. These issues become increasingly difficult to address when considering details related to the organizational aspect of the workflows and security constraints. Among the formal methods which have been proposed for modelling interorganizational workflows, Petri nets present the advantage of an intuitive graphical representation, besides their formal semantics. Their use has been proposed in: [2,3,5,10,12,15,16]. The approaches in [2,3,8,12,18] study a soundness property describing the correct execution

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 486–493, 2012.

of interorganizational workflows, but do not address the organizational dimension of the component workflows and do not impose security constraints. The organizational aspect is considered in [5,17,19]. The correctness problem is only studied in [17], which proposes a nested Petri net model, but does not include security constraints. The importance of considering security constraints for interorganizational workflows has been emphasized in papers like ([6,11,16]), but few approaches consider the verification of the correct execution under security constraints. The approach in [6] uses Petri nets for describing the interorganizational workflow, allows the specification of the organizational aspect and of security constraints, but does not address any correctness issues. In [11], the authors provide access control mechanisms inside the component organizations and use Petri nets for verification, but only in the process dimension. In [16], multilevel security features are included in the interorganizational workflow and the soundness property is discussed, but the approach abstracts from the organizational dimension of workflows.

This paper introduces Nested Interorganizational Workflow Nets (NIWF-nets) - an approach for the modelling of interorganizational workflows which permits the specification of security constraints. In our organizational model, permissions to execute certain tasks are associated to roles, where a role represents a class of users with similar capabilities. In order to increase flexibility and efficiency, we allow certain roles to be used by external organizations for the execution of tasks. Our model permits the enforcing of security constraints: we consider separation of duties constraints (SOD), which have been identified as an efficient mechanism to prevent fraud inside an organization ([4,7,13]). Such constraints can specify that a user who executed a task in the current workflow instance, cannot execute a set of other tasks, in order to prevent fraudulent actions of the user. SOD constraints are particularly useful when certain roles are involved in executing tasks from different organizations, as they can be used to avoid conflict of interests and undesired access to sensitive information.

Our approach is based on nested Petri nets ([14]), in which tokens may be Petri nets themseleves (object-nets). In our model, there exist object-nets for every local workflow and every role in each organization. The role-nets will incorporate the mechanisms needed to enforce the SOD constraints. The commnication between component workflows is modelled using a communication net. We will define and analyze a soundness property which describes the correct execution of the interorganizational workflow, under security and resource constraints.

2 Nested Interorganizational Workflow Nets

In this section we propose a model for interorganizational workflows, NIWF-nets (nested interorganizational workflow nets), based on nested Petri nets ([14]).

We will first present an introductory example of our model. We consider an interorganizational workflow which consists of two workflows, WF_1 and WF_2. WF_1 belongs to a company which manufactures products, which are designed by WF_2. The existing roles are: *Analyst* and *Clerk* in WF_1 and *Designer* in

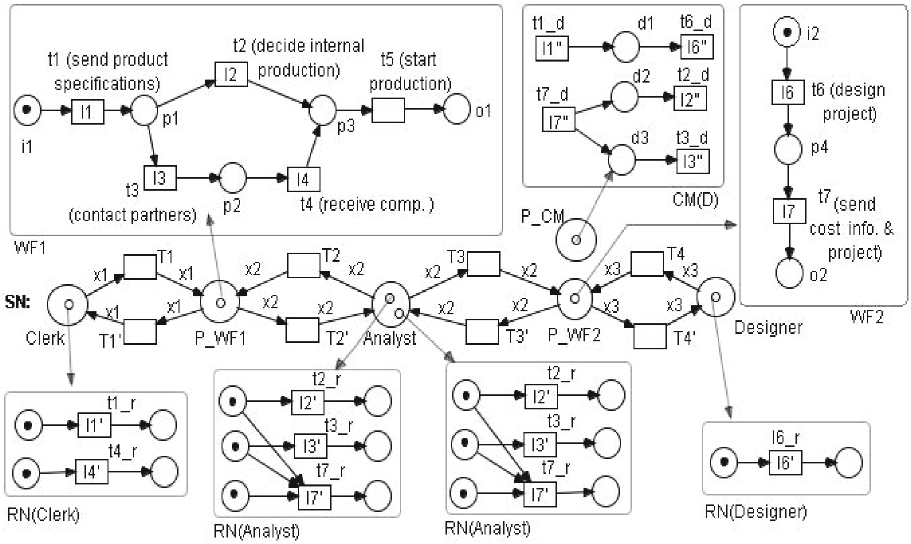


Fig. 1. An example of a NIWF-net modelling an interorganizational workflow

WF_2. In WF_1 (Fig.1) a *Clerk* sends the product specifications to WF_2 (t_1). Upon receiving a request, in WF_2 a *Designer* designs the product (t_6). Then, the cost of the product is established by an *Analyst* and the project is sent to WF_1 (t_7). After receiving the project, in WF_1 an *Analyst* role can decide the internal production (t_2) or that components of the product should be made by external partners (t_3). After the components are received by a *Clerk* (t_4), the production can be started (t_5- does not require a role for its execution). The coordination of the two workflows is described using the dependency relation between tasks: $\mathcal{D} = \{(t_1, t_6), (t_7, t_2), (t_7, t_3)\}$. Thus, task t_6 should fire after t_1, and t_2, t_3 after t_7 in WF_2. In order to avoid conflict of interests, since t_2 and t_3 in WF_1 and t_7 in WF_2 are executed by an *Analyst* role, one should impose that t_2 (t_3) in WF_1 and t_7 in WF_2 should not all be executed by the same user with the *Analyst* role. Thus, the SOD constraints consist of one rule: $t_7 \Rightarrow \neg(t_2, t_3)$.
In the $NIWF$-net in Fig. 1, there exist several object-nets: WF_1 and WF_2 are Petri nets describing the process aspect of the workflows. Given the relation $\mathcal{D}$, the communication object $CM(\mathcal{D})$ contains transitions corresponding to the transitions involved in $\mathcal{D}$. A transition t_d in $CM(\mathcal{D})$ has a label l'' if its corresponding task t has the label l. For every role r, there exists a role-net, $RN(r)$. Since the *Analyst* role can execute t_2, t_3 and t_7, there exist corresponding transitions in $RN(Analyst)$, with labels matching those of t_2, t_3 and t_7 (l'_2, l'_3, l'_7). Because of the SOD rule, t_{7_r} has as input places the input places of t_{2_r} and t_{3_r}. The system net SN consists of two places for WF_1 and WF_2, a place for each role and a place for $CM(\mathcal{D})$. The transitions in SN move the role-nets between the role places and the workflow places. In the marking in Fig.1, t_1 in WF_1 and t_6 in WF_2 are enabled. Because t_6 is labelled and it is involved in $\mathcal{D}$, this means

it can only fire at the same time with t_{6_r} in $RN(Designer)$ and t_{6_d} in $CM(\mathcal{D})$ (i.e. there should be a *Designer* role available at P_{WF_2} and the communication module should permit the firing of this task). Since t_{6_d} is not enabled, t_6 cannot be executed. If T_1 fires in SN, a user with the role $Clerk$ is moved to place P_{WF_1}. Now, a synchronization step (t_1, t_{1_r}, t_{1_d}) is enabled and its firing adds tokens to places p_1, d_1 and to the output place of t_{1_r}. After a role-net $RN(Designer)$ is moved to place P_{WF_2} by firing T_4, the synchronization step (t_6, t_{6_r}, t_{6_d}) is also enabled. Assume that an Analyst role-net has been moved to place P_{WF_2}, the synchronization step (t_7, t_{7_r}, t_{7_d}) fired (i.e an analyst executed t_7) and than the *Analyst* role-net is moved back to place *Analyst*. In the marking of this object-net $ob_2 = (RN(Analyst), m_1)$, there are no tokens in the input places of its transitions. The same analyst cannot execute t_2 in WF_2: if T_2 fires and ob_2 is moved to P_{WF_1}, although t_{2_d} is enabled in $CM(\mathcal{D})$, t_{2_r} is not enabled in ob_2.

In what follows, we will define the specification of an interorganizational workflow, and than, the NIWF-net associated to it.

Each component workflow is executed inside an organization, which must define the roles responsible with the execution of tasks. Several users can be assigned to each role. We permit the existance of external roles: the users assigned to these roles can execute tasks belonging to other workflows. The interaction between the component workflows is specified using a partial relation on those tasks which need to be coordinated. Separation of duties rules can be defined in order to ensure the secure execution of the interorganizational workflow. If a role is authorized to execute several tasks (within or outside the organization), one can restrict a user with this role to execute some possible conflicting tasks.

Thus, a specification $\mathcal{S}$ for an interorganizational workflows is a tuple $\mathcal{S} = < \mathcal{WF}, \mathcal{R}oles, roles, \mathcal{D}, SOD >$ such that:

-$\mathcal{WF} = \{WF_1, \ldots, WF_n\}$ is the set of all the component workflows. Let $\mathcal{T}$ be the set of all the tasks in these workflows.

- $\mathcal{R}oles$ is the set of all the roles in the component workflows.

- $roles : \mathcal{T} \to 2^{\mathcal{R}oles}$ is a function which specifies, for a task, the set of roles which are permitted to execute it.

- the dependency relation $\mathcal{D} \subseteq \mathcal{T} \times \mathcal{T}$ is a partial order which defines the dependency between tasks belonging to different workflows. If $(t_1, t_2) \in \mathcal{D}$, this means that t_1 must execute before t_2.

- SOD is the set of separation of duties rules: a SOD rule $t \to \neg(t_1, \ldots, t_k)$ specifies that, if a certain user with a role r authorized to execute tasks $t, t_1, \ldots, t_k$, executes t, then it cannot execute anylonger none of the tasks $t_1, \ldots, t_k$. Also, if the user executes one of $t_1, \ldots, t_k$, he cannot execute t. If $t \to \neg(t_1, \ldots t_k)$, then it also holds that $t_i \to \neg(t), \forall i \in \{1, \ldots, k\}$. Tasks $t, t_1, \ldots, t_k$ could belong to different workflows. SOD rules can be used to avoid fraudulent actions of a user inside one organization, but also to avoid conflict of interests, if a user executes sensitive tasks in different organizations.

We will use *workflow-nets* (WF-nets [1]) to model the process dimension of workflows. A WF-net contains two special places, i and o, such that $\bullet i = \emptyset$ and $o\bullet = \emptyset$ and every element in the net belongs to a path from i to o. The marking with one token in place i (denoted by i) is the initial marking of the net, while the marking with one token in o (denoted by o) is the final marking.

We will model each $WF_k \in \mathcal{WF}$ by a WF-net. i_k is the initial place of WF_k.

Given a specification $\mathcal{S}$ and a role $r \in \mathcal{R}oles$, the corresponding *role-net* $RN(r) = (P, T, F)$ is obtained as follows:
- $T = \{t_r | \exists t \in \mathcal{T} : r \in roles(t)\}$
- $P = \cup_{t_r \in T} \{in_{t_r}, out_{t_r}\}$
- for each t_r: $(in_{t_r}, t_r), (t_r, out_{t_r}) \in F$; for each t_r such that $t \to \neg(t1, \ldots, tn) \in SOD$, let in_{ti_r} be the input places of the transition ti_r corresponding to ti in $RN(r)$, for all $i \in \{1, \ldots, n\}$. Then, $(in_{t1_r}, t_r), \ldots, (in_{tn_r}, t_r) \in F$

An initial marking for $RN(r)$, denoted by mr_0, contains one token in every input place. For each task that can be executed by a role r, there exists a corresponding transition t_r with one input place and one output place. If there exists a SOD rule involving the tasks which can be executed by r, $t \to \neg(t1, \ldots, tn)$, then let t_r, $t1_r, \ldots tn_r$ be the transitions corresponding to the tasks in the rule. The input places of $t1_r, \ldots tn_r$ are connected to t_r, hence the firing of t_r prevents $t1_r, \ldots tn_r$ from firing. Also, if any transition ti_r fires, it disables t_r, which is consistent with the fact that t_i and t_r cannot be executed by the same role. Thus, SOD constraints will be enforced through the structure of $RN(r)$.

Given a specification $\mathcal{S}$, a communication net associated to the dependency relation $\mathcal{D}$, $CM(\mathcal{D}) = (P_D, T_D, F_D)$, is defined as follows:
- $P_D = \{p_d | d \in \mathcal{D}\}$: a place for each pair of dependent tasks.
- $T_D = \{t_d | \exists (t', t) \in \mathcal{D} \vee (t, t') \in \mathcal{D}\}$: a transition for each task involved in $\mathcal{D}$.
- $F_D = \{(p_d, t_d) \in P_D \times T_D | d = (t', t) \in \mathcal{D}\} \cup \{(t_d, p_d) \in T_D \times P_D | d = (t, t') \in \mathcal{D}\}$

An initial marking for $CM(\mathcal{D})$, denoted by md_0, is the empty marking.

Definition 1. *Given a specification* $< \mathcal{WF}, \mathcal{R}oles, roles, \mathcal{D}, SOD >$*, a nested interorganizational workflow net,* $NIWF$*, is a tuple*
$NIWF = (Var, WF\_nets, Role\_nets, CM(\mathcal{D}), SN, L, \lambda)$ *such that:*

- $Var = \{x_r | r \in \mathcal{R}oles\}$ *is the set of variables in the net.*
- $WF\_nets = \{WF_1, \ldots, WF_n\}$ *is set of WF-nets corresponding to* $\mathcal{WF}$*.*
- $Role\_nets = \{RN(r) | r \in \mathcal{R}oles\}$ *is a set of role-nets.*
- $CM(\mathcal{D})$ *is an object-net (communication net), obtained from relation* $\mathcal{D}$*.*
- $SN = (P_{SN}, T_{SN}, F_{SN}, M_0)$ *is the system net:*
  - P_{SN} *contains a workflow place* P_{WF} *for each workflow* WF*, a role place* P_r *for each role* $r \in \mathcal{R}oles$ *and a place* P_{CM}*.*
  - T_{SN} *and* F_{SN} *contain transitions and labelled arcs connecting role places to workflow places: Let* P_r *be a role place. For each workflow* WF_i *in which* r *can execute tasks,* $\exists T_i, T_i' \in T_{SN}$ *such that* (P_r, T_i)*,* (T_i, P_{WFi})*,* (P_{WFi}, T_i')*,* (T_i', P_r)*,* $\in F_{SN}$ *and all these arcs are labelled with the variable* x_r*.*
  -M_0 *assigns marked object-nets to the places of* SN*:* $M_0(P_{WFk}) = (WF_k, i_k)$*, for all* $k \in \{1, \ldots, n\}$*,* $M_0(P_r) = n_r'(RN(r), mr_0)$ *(*$n_r \in \mathbb{N}$*) for each* $r \in \mathcal{R}oles$ *(i.e. there are* n_r *role-nets* $RN(r)$ *in* P_r*) and* $M_0(P_{CM}) = (CM(\mathcal{D}), md_0)$*.*
- $L = L_{WF} \cup L_R \cup L_D$ *is a set of labels:* L_{WF} *is used for transitions in the component workflows;* $L_R = \{l' | l \in L_{WF}\}$ *;* $L_D \subseteq \{l'' | l \in L_{WF}\}$*.*

– *λ is a partial labelling function on transitions, such that:*
  - tasks in the workflows can have a label from L_{WF}
  - for each $t \in \mathcal{T}$, $r \in Role(t)$, let t_r be the transition corresponding to t in $RN(r)$. Then, $\lambda(t_r) = l'$, where $l = \lambda(t)$
  - if $t \in \mathcal{T}$ such that $(t, t') \in \mathcal{D}$ or $(t', t) \in \mathcal{D}$, then for $t_d \in T_D : \lambda(t_d) = l''$, where $l = \Lambda(t) \in L_{WF}$.

A net-token is a marked object-net. Let A_{net} be the set of all net tokens.
A marking of $NIWF$ is a function M which assigns to each place P a multiset of net-tokens: $M(P) : A_{net} \to \mathbb{N}$.

Let T a transition in SN, P its input place and P' its output place. Let $P_r \in \{P, P'\}$ be the corresponding role place connected to it and $var(T) = x_r$, where x_r is the variable labelling the arc between P_r and T.
- A binding b associated to T assigns to x_r a net-token $(RN(r), m)$.
- If M is a marking of a NIWF-net, T *is enabled in M w.r.t. a binding* b if $b(var(T)) \in M(P)$. The firing of T produces a new marking, M': $M[T[b]\rangle M'$, such that: $M'(P) = M(P) - b(var(T))$ and $M'(P') = M(P') + b(var(T))$.

There are several types of steps, defining the behaviour of NIWF-nets:
(1)**Transport step:** If T is a transition in SN enabled in a marking M w.rt. a binding b, then T is a transport step.
(2)**Object - autonomous step:** Let M be a marking, P a place of SN, $\alpha = (EN, m) \in M(P)$ and t a transition in α such that $\lambda(t)$ is undefined. t is an object-autonomous step enabled in M if $m[t\rangle m'$. The resulting marking of $NIWF$, M', is obtained from M by replacing, in $M(P)$, the net-token α with the net-token $\alpha' = (EN, m')$.
(3)**Synchronization step:** Let M be a marking, P a place of SN. Let $\alpha_1 = (EN_1, m_1)$ and $\alpha_2 = (EN_2, m_2)$ two net-tokens in $M(P)$ such that (1) $\exists\ t$ in (EN_1, m_1) with $\lambda(t) = l \in L_{WF}$ and $m_1[t\rangle m_1'$; (2) $\exists\ t'$ in (EN_2, m_2) with $\lambda(t') = l' \in L_R$ and $m_2[t'\rangle m_2'$; (3) if $\exists\ t''$ in $(CM(\mathcal{D}), m)$ with $\lambda(t'') = l'' \in L_D$, then $m[t''\rangle m'$. Then, we say that $Y = (t, t', t'')$ (or $Y = (t, t')$, if t'' does not exist) represents a synchronization step enabled in marking M. The resulting marking, M', is obtained from M by replacing in $M(P)$ the net-token α_1 with the net-token $\alpha_1' = (EN_1, m_1')$, the net-token α_2 with the net-token $\alpha_2' = (EN_2, m_2')$ and $(CM(\mathcal{D}), m)$ with $(CM(\mathcal{D}), m')$ in $M(P_{CM})$.

If t in α_1 has a label $l \in L_{WF}$, then α_1 is a WF-net. If t' in α_2 has a label l', then α_2 is a role-net and t' is a transition t_r corresponding to t. If there exists t'' in $CM(\mathcal{D})$ with a label l'' (i.e. $t'' = t_d$ and t is involved in external interaction with transitions from other workflows) t'' should be enabled and t, t' and t'' must fire synchronously, that is: a user assigned to role r, when executing task t, interacts with the communication module (sends a message) in order to check if the external conditions for executing the task are met.

3 Verification of Interorganizational Workflows

A notion of soundness was defined for WF-nets, expressing the minimal conditions a correct workflow should satisfy ([1]). A WF-net is sound iff (1) for every marking m reachable from marking i, there exists a firing sequence from m to marking o : $(\forall m)((i[*\rangle m) \Longrightarrow (m[*\rangle o))$; (2) all the transitions in WF can fire.

A final marking of a NIWF-net is a marking in which all the object role-nets reside in their initial places and each workflow place only contains the corresponding workflow, in a final marking. We denote by $\mathcal{M}_f$ the set of final markings. A marking in the set $\mathcal{M}_f$ is reachable iff all the component workflows successfully ended their execution, even when their behaviour is restricted by the SOD and resource constraints and by the communication mechanisms.

Definition 2. *A nested interorganizational workflow net $NIWF$ is sound iff:*
(1) (WF_j, i_j) is a sound workflow net, $\forall j \in \{1, \ldots, n\}$.
(2) Every step Y should be able to fire: $(\exists M \in [M_0\rangle : M[Y\rangle)$.
(3) For every reachable marking M, there exists a firing sequence to a marking from $\mathcal{M}_f$: $(\forall M)((M_0[*\rangle M) \Longrightarrow (M[*\rangle M_f, M_f \in \mathcal{M}_f))$.
(4) The communication net is bounded: $(\exists n \in N)(\forall M \in [M_0\rangle$, $(CM(\mathcal{D}), m) \in M(P_{CM}))(\forall p_d \in P_D : m(p_d) \leq n)$.

Condition (4) requires that the participant workflows should not be allowed to send an infinite number of messages to the communication module.

Using the results from [9,14], it can be proven that the boundedness problem is decidable for NIWF-nets. Also, it holds:

Lemma 1. *Assume $NIWF$ is a NIWF-net, such that all the component WF-nets are sound. Then, $NIWF$ is bounded iff the communication net is bounded.*

Theorem 1. *The soundness is decidable for NIWF-nets.*

Condition (1) in Def. 2 is decidable ([1]). Condition (2) is also decidable (it is decidable for general nested Petri nets: [14]). The boundness of $NIWF$ is decidable. If (1) holds and $NIWF$ is unbounded, then condition (4) does not hold (Theorem 1), so the NIWF-net is not sound. If $NIWF$ is bounded, then (4) holds and $[M_0\rangle$ is finite. Since $\mathcal{M}_f$ is finite, condition (3) is also decidable.

4 Conclusions and Future Work

In this paper we introduced a Petri net model, NIWF-nets, for interorganizational workflows which permits the specification of roles in the component workflows and the mobility and sharing of resources across organizations. Our model allows the specification of security constraints and offers a modular view of the interorganizational workflow. We defined a soundness property for NIWF-nets and proved its decidability. Our future work aims at extending the organizational model and including other security constraints.

References

1. van der Aalst, W.: The Application of Petri nets to Workflow Management. The Journal of Circuits, Systems and Computers 8(1), 21–66 (1998)

2. van der Aalst, W.: Loosely coupled interorganizational workflows: Modeling and analyzing workflows crossing organizational boundaries. Information and Management 37(2), 67–75 (2000)
3. van der Aalst, W.M.P., Lohmann, N., Massuthe, P., Stahl, C., Wolf, K.: From Public Views to Private Views – Correctness-by-Design for Services. In: Dumas, M., Heckel, R. (eds.) WS-FM 2007. LNCS, vol. 4937, pp. 139–153. Springer, Heidelberg (2008)
4. Ahn, G., Sandhu, R.: The RSL99 Language for Role-based Separation of Duty Constraints. In: 4th ACM Workshop on Role-Based Access Control, pp. 43–54. ACM, New York (1999)
5. Andonoff, E., Bouaziz, W., Hanachi, C.: An Agent-Based Model for Autonomic Coordination of Inter-Organizational Business Processes. INFORMATICA 20(3), 323–342 (2009)
6. Ayed, S., Cuppens-Boulahia, N., Cuppens, F.: Deploying Access Control in Distributed Workflow. In: The 6th Australasian Conference on Information Security, pp. 9–17. Australian Computer Society Inc., Darlinghurst (2008)
7. Bertino, E., Ferrari, E., Atluri, V.: The Specification and Enforcement of Authorization Constraints in Workflow Management Systems. ACM Trans. on Inf. and Sys. Sec. 2(1), 65–104 (1999)
8. Ge, J., Hu, H., Lu, P., Hu, H., Lü, J.: Translation of Nets Within Nets in Cross-organizational Software Process Modeling. In: Li, M., Boehm, B., Osterweil, L.J. (eds.) SPW 2005. LNCS, vol. 3840, pp. 360–375. Springer, Heidelberg (2006)
9. Finkel, A., Schnoebelen, P.: Fundamental Structures in Well-Structured Infinite Transition Systems. In: Lucchesi, C.L., Moura, A.V. (eds.) LATIN 1998. LNCS, vol. 1380, pp. 102–118. Springer, Heidelberg (1998)
10. Jiang, P., Shao, X., Gao, L., Qiu, H., Li, P.: A process-view approach for cross-organizational workflows management. Adv. Eng. Inf. 20(2), 229–240 (2010)
11. Kang, M.H., Park, J.S., Froscher, J.N.: Access control mechanisms for inter-organizational workflows. In: The 6th ACM Symposium on Access Control Models and Technologies, pp. 66–74. ACM, New York (2001)
12. Kindler, E., Martens, A., Reisig, W.: Inter-operability of Workflow Applications: Local Criteria for Global Soundness. In: van der Aalst, W., Desel, J., Oberweis, A. (eds.) BPM 2000. LNCS, vol. 1806, pp. 235–253. Springer, Heidelberg (2000)
13. Knorr, K., Weidner, H.: Analyzing Separation of Duties in Petri Net Workflows. In: Gorodetski, V.I., Skormin, V.A., Popyack, L.J. (eds.) MMM-ACNS 2001. LNCS, vol. 2052, pp. 102–114. Springer, Heidelberg (2001)
14. Lomazova, I.A.: Nested Petri Nets - a Formalism for Specification and Verification of Multi - Agent Distributed Systems. Fund. Informa. 43, 195–214 (2000)
15. Lohmann, N., Massuthe, P., Stahl, C., Weinberg, D.: Analyzing Interacting WS-BPEL Processes Using Flexible Model Generation. Data Knowl. Eng. 64(1), 38–54 (2008)
16. Mikolajczak, B., Gami, N.: Design and Verification of Loosely Coupled Inter-Organizational Workflows with Multi-Level Security. JCP 3(1), 63–78 (2008)
17. Prisecaru, O.: A Petri net based approach for modelling resource constrained interorganizational workflows. In: Ultes-Nitsche, U., Moldt, D., Augusto, J.C. (eds.) MSVVEIS 2008, pp. 29–38. Insticc Press (2008)
18. Yamaguchi, S., Matsuo, H., Ge, Q., Tanaka, M.: WF-Net Based Modeling and Soundness Verification of Interworkflows. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. E90-A(4), 829–835 (2007)
19. Wirtz, G., Giese, H.: Using UML and object-coordination-nets for workflow specification. In: 2000 IEEE International Conference on Systems Man and Cybernetics Cybernetics, pp. 3159–3164. IEEE Press (2000)

Towards Specialization of the Contract-Aware Software Development Process

Anna Derezińska and Przemysław Ołtarzewski

Institute of Computer Science, Warsaw University of Technology,
Nowowiejska 15/19, 00-665 Warsaw, Poland
A.Derezinska @ii.pw.edu.pl

Abstract. The contract-based software implementation improves accuracy and verification capabilities of business information systems. This paper promotes contract identification in early phases of the software development and defining contracts within models. Design and implementation artifacts that are responsible for system functionality and system constraints are transformed during the consecutive development phases. Combination of the Model Driven Engineering paradigm and Design by Contract ideas constitutes the Contract Aware Software Development process (CASD). The approach is specialized for system modeling in UML, contracts specified at model level in Object Constraint Language (OCL), and the final implementation in the C# language. The specialized process is supported by the tool transforming the models with associated contracts into the corresponding programs with contracts at the code level.

Keywords: software development process, Model Driven Engineering, UML, contracts, OCL.

1 Introduction

An apparent discrepancy between a system specification and its implementation is a common problem arisen in business systems. A tendency to overlook system constraints can be especially observed during requirement analysis and software modeling and design. The constraints are often modified, as in the evolution of enterprise systems. These error-prone phenomena result in increase of the number of system faults and high costs of a software correction and maintenance.

The answer to these problems lays in using contracts through the whole software development process, including especially modeling activities. Contracts can be understood in general as conditions that must be true for the whole life time of a system unit, or be satisfied at certain time points of a unit behavior. A unit is represented by different kinds of models or software descriptions.

Design-by-contract (DbC^{TM}) [1] is a technique that relies on the specification of contracts and is primarily used at the code level. However, the code contracts originate from constraints specified in business rules and software requirements. Consequently the contracts could already be expressed in models at the system design level.

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 494–501, 2012.

This paper is focused on the application of the contract-based methodology combined with model-driven engineering [2]. We propose a general approach, so-called Contract-Aware Software Development (CASD) process [3], in which business system constrains are specified as contracts and applied to system models. Next, the models with associated contracts can be refined and transformed into the code level.

This generic approach can be specialized towards different modeling languages, e.g. UML, and various contract technologies. A language used for formulating of constraints in UML models is the Object Constraint Language (OCL) [4,5]. However, in order to achieve the anticipated objectives it is indispensable to provide an infrastructure supporting the approach. Contract to code transformation is integral to the way the methodology is applied. The contract implementation effort is moved to the discussed transformation tool. Therefore, the code is consistent with the specification. Moreover, the labor required for the contract maintenance can be lowered.

In the next Section, the main issues of the Contract-Aware Software Development process are specified. Section 3 presents a solution adapting the process to selected technologies. In further sections we discuss how some ideas of the CASD process are applied with the tool support and related to other work. Section 6 concludes the paper.

2 Contract-Aware Software Development

Contract-Aware Software Development (CASD) is an approach to a generic process that combines features of the contract-based and model-driven development. The preliminary draft of the process was presented in [3,6]. The main phases of the process and its basic artifacts are illustrated in Fig. 1.

Dual artifacts encounter in various phases of the process, from the analysis to the implementation one. The artifacts shown on the left hand side correspond to a system functionality, whereas the opposite elements represent constraints and contracts.

The artifacts within a development phase are related by a *constrain* dependency. It represents the coupling between the system functionality and the corresponding contract at the same abstraction level. The inter-phase relations are defined by *transform* and *trace* dependencies. *Transform* dependencies illustrate the gradual refinement of the artifacts during the development process. The latter dependencies are responsible for defining traceability to the original artifacts. During the system development and maintenance, changes in a contract made at the i-th level of abstraction should also be incorporated to the previous level. This fact denotes the need for the round trip engineering, which refers to the contracts as well as to the functional artifacts. Both types of the artifacts should be transferred into the next abstract level in a common step.

In general, the basic concepts of the process are defined as a finite sequence $< L_1, \delta_1, L_2, \ldots \delta_n, L_{n+1} >$ including two kinds of elements. The first kind of elements $L_i = (F_i, C_i, \lambda_i)$ is a tuple denoting an i^{th} level of a system abstraction, where:

i = 1…n+1 is a number of a considered level of system abstraction,

i^{th} level is more abstract than $(i+1)^{th}$ level,

F_i is a finite set of functionality artifacts at the i^{th} level of system abstraction,

C_i is a finite set of contracts at the i[th] level of system abstraction,
$\lambda_i : C_i \to F_i$ is a function representing the *constrain* dependencies.

The second kind of elements in the process is a transformation function between two adjacent levels of system abstraction: $\delta_i : 2^{F_i} \times 2^{C_i} \to 2^{F_{i+1}} \times 2^{C_{i+1}}$, where 2^X denotes a power set of set X and i =1…n.

The transformation function satisfies the following implication:

$$\forall c_i \{((\lambda_i(c_i) = f_i) \wedge \exists \delta_i((\delta_i(A_i, B_i) = (A_{i+1}, B_{i+1})) \wedge (f_i \in A_i) \wedge (c_i \in B_i))) \quad (1)$$
$$\Rightarrow \exists \lambda_{i+1}((\lambda_{i+1}(c_{i+1}) = f_{i+1}) \wedge (f_{i+1} \in A_{i+1}) \wedge (c_{i+1} \in B_{i+1}))\}$$

where:

$f_i \in F_i, f_{i+1} \in F_{i+1}$ are functionality artifacts from the i[th] and i+1[th] levels,
$c_i \in C_i, c_{i+1} \in C_{i+1}$ are contracts from the i[th] and i+1[th] levels,

$A_i \subseteq 2^{F_i}, A_{i+1} \subseteq 2^{F_{i+1}}, B_i \subseteq 2^{C_i}, B_{i+1} \subseteq 2^{C_{i+1}}$ are subsets of functionality artifacts and contracts from the i[th] and i+1[th] levels, respectively.

The formula (1) depicts preservation of *constrain* dependencies between the corresponding artifacts in a transformation. If a pair of artifacts is related at i[th] level, then after transformation, their resulting artifacts at (i+1)[th] level should also be related.

The identified phases do not suggest that the process follows the rules of the waterfall process. The succession of phases shown in Fig. 1 focuses on the main idea recognizing the fundamental artifacts of the process. Possible back dependencies and the overlapping of phases are omitted in the figure.

The dependencies within the process, as well as the automated transformation of functionality artifacts and contracts support system evolution, especially for changing business requirements. This corresponds to the intrinsic postulate of the agile approaches and enterprise system development. In CASD, changes in the functionality and/or in constraints can be propagated through the appropriate process levels.

3 CASD Specialization for UML, OCL and C#

The general idea of the generic CASD process can be specialized by applying various methods and techniques for selected artifacts and their transformations.

The definition and maintenance of the *constrain* dependencies (Fig.1) is important for the process realization. It is recommended to have a tight coupling between artifacts of the both sides of the process, preferably tool supported. Otherwise the additional overhead and manual effort would discourage the concurrent development of both types of the artifacts. Moreover, certain contracts might be easily missed due to an erroneous omission by a developer. Another recommendation is automating of the intra-phase transitions and maintenance of the trace dependencies.

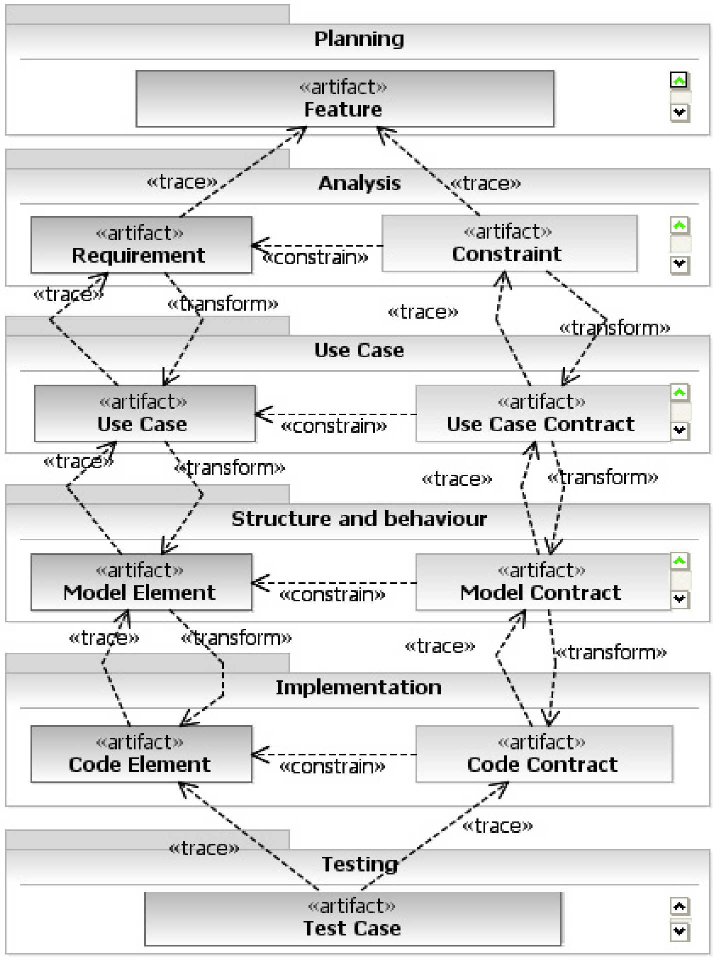


Fig. 1. The Contract Aware Software Development process

System models can be designed in UML, especially class diagrams accompanied by logical constraints specifying business contracts at the model levels [7]. They can be specified in OCL [4,5]. OCL is a declarative specification language. OCL expressions can be used for defining class invariants, pre- and post-conditions of operations, and other constraints associated with model artifacts.

UML is a language that can be applied at different levels of abstraction and used for different purposes. Model refinement towards a selected Domain Language and desired technology can be accomplished by application of model profiles.

Apart from model to model transformations aimed at the gradual model refinement, models can be translated into the source code, or at least to a correct but incomplete subset of the final code. One of many possible solutions, which can be used for the implementation of business systems, is the C# programming language.

A crucial part of such a specialized CASD process is refinement of UML models and transformation of OCL into contracts at the code level. The C# language does not directly include concepts of contracts, but we can utilize the Microsoft Code Contract library [9] that implements contracts. The library integrates with the .NET 4.0 platform and supports C# and Visual Basic programming languages.

The basic part of the library is the *Contract* class including a set of static methods that enable description of contracts. Invariants can be specified for classes, structs and interfaces. Pre- and post-conditions can be associated with constructors, methods, overloaded operators, type conversions and accessor methods of properties, events and indexers. A special static event defined by the *Contract* class is called in case of contract invalidation. Transformation of OCL invariants of classes and pre-, post-conditions specifying operations can be based on the *delegate* concept of C#. The details of this transformation are omitted due to brevity reasons.

4 Transform OCL Fragments into C# - T.O.F.I.C Tool

The core part of the CASD process constitutes the transformation of models and their contracts into the executable applications including the corresponding contracts. The transformation should satisfy preservation of dependencies defined in Sec. 2. To put this into practice the transformation should be automated and the modeling activities assisted in a friendly manner.

The specialized process can be supported by the T.O.F.I.C. tool (*Transform OCL Fragments into C#*) [6]. This tool extends the CASE tool - the IBM Rational Software Architect [10] with the C# modeling and code generation capabilities. The tool consists of a set of plug-ins of the Eclipse framework that creates so-called Eclipse *feature*. The main characteristics of the preliminary prototype version of the tool were described in [11].

The current version of T.O.F.I.C. 1.1.7 was considerably extended in comparison to its prototype. It covers new functionality (e.g. most of OCL, many structures of C#, contracts in C#) and improves its ergonomic features.

The UML profiles are used for modeling of Domain-Specific Languages (DSL), in this case C# concepts and code mapping units. The approach could be adjusted to the modeling of various business information systems according to their needs.

The current version of T.O.F.I.C. was enhanced with the profile tooling. It includes GUI elements that support visualization and editing of the stereotyped elements of a model. Rapid prototyping and assigning of stereotypes to the selected model elements is realized by the C# Action Tool (CAT). It extends the palette of modeling menus with the appropriate view. Various buttons with graphical markers can be used to assign a selected stereotype and visualizes this distinction in the diagram.

The tool also facilitates the C# code generation from a refined UML model and OCL constraints. The generated code is extended with the C# implementation of the standard OCL library. Transformation of OCL contracts is realized using the Visitor design pattern. The code generation is initiated with the creation of the Abstract Syntax Tree (AST) of an OCL expression based on its text representation in a model.

Next, according to the appropriate transformation rule, the nodes of the tree are visited in a given order. During the tree traversal the corresponding C# code is generated and stored in the related compilation unit.

Another new feature, the most significant in the context of the CASD process, is usage of contracts in the target code of OCL. Expressions of OCL are translated into the corresponding method calls of the Microsoft Code Contracts library [9].

Application of the library benefits from an existing contract solution that integrates with the Visual Studio - the commonly used development framework. However, it has also negative consequences due to limitations of the library.

The .NET 4.0 platform supports defining of contracts and application of the dedicated namespace. Verification of contracts is realized by the additional library that should be installed within the Microsoft Visual Studio environment. The library is supported by the tools available via a command line as well as using a GUI extension. Contracts are verified during a project building and in the runtime.

A limitation of the library concerns reaction to a contract invalidation. A default reaction on a raised exception can be modified by a developer by implementing the own class for handling contracts in the runtime. Furthermore, there is no information which instance is responsible for a contract invalidation. Therefore the exception triggers handling of all delegates associated with the exception.

The usability of the approach, the maturity of the extended tool and the impact of adopting contract code generation in the software development process were evaluated in an experiment. The experiment was conducted by students of an advanced course of software engineering. The participants of the experiment cooperated on the development of a common system, eliciting and specifying requirements of particular system modules. The system simulated business and control processes of an airport. The requirement specifications took into account various system constraints. Next, the constraints were included in the description of use cases.

The team members swap the requirement specifications among others. UML models were designed according to the obtained specifications. The general models were supplemented with appropriate contracts written in OCL and refined to the C# code models with assistance of CAT. Then, the C# projects were generated from the refined models by the T.O.F.I.C. tool. The OCL contracts were transformed to the corresponding code using the Microsoft Code Contracts library.

In the experiment the impact of the automatic code generation from models with contracts to the software development was examined. The results of the experiment were evaluated in two ways. The models, other intermediate artifacts, and final applications with tests were examined by hand and using static and dynamic verification tools. In addition, the participants filled in a questionnaire after the experiment.

The general evaluation of all results confirmed the improved consistency between the code and this level of specification that was expressed in refined models and OCL constraints. In the questionnaire, the most of participants admitted that they would be utilizing the T.O.F.I.C. tool to a project development in the future, assuming the selected improvements were incorporated.

The key obstacles of the approach are problems of consistency between preliminary business rules described in requirements and use cases on the one hand and contracts at the model level on the other hand, the increased effort required during refinement of models, and the obligation of knowledge of OCL.

Another alternative to the process specialization could be combining UML models with constraints written in a target implementation language, e.g. Java or C#. This kind of specification of business rules requires validation of constraints at the model level, in order to be of any practical use.

5 Related Work

Design by Contract principles [1] are first of all applied at the code level, as in the Eiffel language [12]. However, there is a lack of tools dealing with contracts on both abstract levels, as models and code.

There are many tools that support OCL [13,14], but the most of them do not generate code from OCL constraints. In [15] such tools were compared taking into account their contract generation capabilities. All these tools, apart from T.O.F.I.C., generate Java code from OCL. There was announced a vision of a potential, preferred OCL tool, but it is still a future work [16].

Using Java as a target language, we can utilize Dresden OCL [17] in a specialized process with OCL. This tool transforms OCL expressions into aspects of AspectJ. OCL constraints used in MDE towards Corba and Java can also be found in [18,19].

Support for the C# code generation has been incorporated into several CASE tools, but none of them supports contract-based approach for C# at the generated code level.

6 Conclusions

The paper presents an approach that combines model-driven development principles with advantages of contract utilization. The approach was applied using selected modeling, specification and implementation languages. The early definition of contracts, as recommended in the CASD process, focuses the attention of developers on the constrains and their verification. They should reflect business rules of a system. OCL constraints are transformed to code contracts supported by a library in order to move the benefits of design by contract approach to the modeling level.

The preliminary experiment concludes that the tool supported contract evaluation combined with the model-driven methodology could improve the application accuracy and testability. The applicability of the methodology depends strongly on the convenient tool support, which has to be further enhanced. The critical issues also remain completeness of system constraints implemented as contracts.

As far as the obstacles in the contract utilization with T.O.F.I.C. are concerned, the contract library used in the tool could be substituted by another contract solution, or there will be available an improved version of the Microsoft Code Contracts library.

The solution can be extended with code generation from the dynamic models, e.g. state machines. State invariants and guard conditions can also be transformed into the appropriate code contracts.

References

1. Meyer, B.: Object-Oriented Software Construction, 2nd edn. Prentice-Hall (1997)
2. France, R., Rumpe, B.: Model-driven Development of Complex Software: A Research Roadmap. In: Proc. of Future of Software Engineering at ICSE 2007, pp. 37–54. IEEE Soc. (2007)
3. Derezińska, A., Ołtarzewski, P.: Business Software Development Process Combining Model-Driven and Contract-Based Approaches. In: Jałowiecki, P., Łuasiewicz, P., Orłowski, A. (eds.) Information Systems in Management XI, pp. 7–17. WULS Press, Warsaw (2011)
4. Object Constraint Language (OCL) (March 15, 2012), `http://www.omg.org/spec/OCL/`
5. Warmer, J., Kleppe, A.: The Object Constraint Language: Getting Your Models Ready for MDA. Addison-Wesley Professional (2008)
6. Ołtrzewski, P.: Software Development Using Contracts by Example of the T.O.F.I.C Tool. Master Thesis, Inst. of Computer Science, Warsaw Univ. of Technology (2011) (in Polish)
7. Neumarite, L., Ceponiene, L., Vadrickas, G.: Representation of Business Rules in UML&OCL Models for Developing Information Systems. In: Stirna, J., Persson, A. (eds.) POEM 2008. LNBIP, vol. 15, pp. 182–196. Springer (2008)
8. Frankel, S.: Model Driven Architecture: Appling MDA to Enterprise Computing. Wiley Press, Hoboken (2003)
9. Microsoft Code Contracts (March 15, 2012), `http://research.microsoft.com/en-us/projects/contracts`
10. IBM Rational Software Architect (March 15, 2012), `http://www-01.ibm.com/software/awdtools/swarchitect/`
11. Derezińska, A., Ołtarzewski, P.: Model-Driven Engineering Support for Building C# Applications. In: Sobh, T., Elleithy, K. (eds.) Innovations in Computing Sciences and Software Engineering, pp. 449–454. Springer (2010)
12. Eiffel Software: An Eiffel Tutorial (March 15, 2012), `http://docs.eiffel.com/`
13. Chimiak-Opoka, J., Demuth, B., et al.: OCL Tools Report Based on the IDE4OCL Feature Model. In: Proc. of International Workshop on OCL and Textual Modeling, col. Tools Europe (2011)
14. Toval, A., Requena, V., Fernandez, J.L.: Emerging OCL tools. Journal of Software and System Modeling 2(4), 248–261 (2003)
15. Derezińska, A., Ołtarzewski, P.: Code Generation of Contracts Using OCL Tools. In: Borzemski, L., et al. (eds.) Information Systems Architecture and Technology, Web Information Systems Engineering, Knowledge Discovery and Hybrid Computing, pp. 235–244. Publishing House of Wroclaw University of Technology, Poland (2011)
16. Chimiak-Opoka, J., Demuth, B.: A Feature Model for an IDE4OCL. In: Proc. of International Workshop on OCL and Textual Modeling (2010)
17. Dresden OCL, (March 15, 2012), `http://reuseware.org/index.php/DresdenOCL`
18. Coronato, A., De Pietro, G.: Formal design and implementation of constraints in software components. Advances in Engineering Software 41, 737–747 (2010)
19. Dan, L., Danning, L.: Applying Model Driven to Software Development: a University Library Case Study. In: Proc. of the 3rd International Conference on Communication Software and Networks, ICCSN, pp. 179–183. IEEE Comp. Soc. (2011)

Branching Processes Theory Application for Cloud Computing Demand Modeling Based on Traffic Prediction

Victor Romanov, Aleksandra Varfolomeeva, and Andrey Koryakovskiy

Department of Computer Science, Russian Plekhanov University of Economics, Moscow, Russian Federation
{victorromanov1,aovarfolomeeva}@gmail.com, avkor@list.ru

Abstract. With cloud computing growing in popularity cloud-service providers must guarantee that data are processed rapidly and transferred when and where they are needed. Unfortunately, it is extremely difficult to predict the exact performance characteristics and demands on the network at any particular time. In this paper we show that the cloud computing demand can be developed as a branching stochastic process. Branching processes are used to describe random systems such as population development, nuclear chain reactions and spread of epidemic disease. A statistical model is described and using this model we propose a method for determining the unknown probability distribution of queries. Network traffic modeling is an issue of great importance to both consumers and providers of cloud-based services. Firstly, traffic modeling helps to represent our understanding of dynamic demand for cloud services by stochastic processes. Secondly, accurate traffic models are necessary for service providers to properly maintain quality of service.

Keywords: cloud computing, branching process, network traffic modeling.

1 Introduction

Cloud computing is one of the hottest topics in all of information technology today. This is a new model of delivering computing resources in which centrally administered computing capabilities are provided as services on-demand over the network to a variety of customers. According to IDC's analysis, the worldwide forecast for cloud services in 2013 will amount to $44.2bn, with the European market reaching €6,005m in 2013.[1] The potential benefits of cloud computing like lower costs, faster implementation, and more flexibility are overwhelming. However, attaining these benefits requires that new technologies and solutions managing the huge number of operations and volumes of data within a cloud transparently and without service interruptions emerge. As cloud computing enables users to store all their data on the network predicting net-

[1] IDC *Cloud Computing 2010 - An IDC Update,* Frank Gens, Robert P Mahowald, Richard L Villars, Sep 2009 - Doc # TB20090929, 2009.

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 502–510, 2012.

work performance and cloud-based services demand may become a challenging issue. In this paper we suggest that branching processes theory will fit for both describing the dynamics of cloud services demand and predicting the network traffic in cloud computing environment. We consider that more and more clients will know about cloud services, one client from another, like epidemic of disease spreads. We briefly describe the basic principles of cloud computing concept and introduce branching processes theory. The purpose of this paper is to provide a stochastic model that would be helpful for both consumers and providers of cloud-based services. On the one hand, traffic modeling helps to represent our understanding of dynamic demand for cloud services by stochastic processes, and on the other hand accurate traffic models are necessary for service providers to avoid "bottlenecks" and to improve quality of services provided.

2 Literature Review

The design of robust and reliable network services for cloud computing environment is a challenging task. The only path to achieve this goal is to develop a detailed understanding of the traffic characteristics. An accurate estimation of the network performance is vital. Traffic models enable network designers to make assumptions about the networks being designed based on past experience and also enable prediction of performance for future rapidly changing requirements in cloud environment [10]. A corpus of literature on network traffic modeling exists. One of the most widely used and oldest traffic models is the Poisson Model. The Poisson process is characterized as a renewal process and Poisson distribution is the predominant model used for analyzing traffic in traditional telephony networks [11]. Deterministic Traffic Model [12] is proposed for providing real time service over real time channel where clients declare their traffic characteristics and performance requirement at the time of channel establishment in this model. Chaotic maps are low dimensional nonlinear systems whose time evolution is described by knowledge of an initial state and a set of dynamical laws. In [13] the author illustrates traffic characteristics that can be modeled by considering chaotic maps. Wavelet-based models use wavelet transform function to model long-range dependence traffic such as traffic measured on Ethernet. Multifractal wavelet model is presented in [14].

3 Cloud Computing

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics: on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service.[2]

[2] NIST: Cloud Computing Program: http://www.nist.gov/itl/cloud/index.cfm

There are three service models of cloud computing:

Software as a service (SaaS): is software offered by a third party provider, available on demand, usually via the Internet configurable remotely. Examples include online word processing and spreadsheet tools, CRM services (Salesforce CRM [5], Google Docs).
Platform as a service (PaaS): allows customers to develop new applications using APIs deployed and configurable remotely. The platforms offered include development tools, configuration management, and deployment platforms. Examples are Microsoft Azure [3] and Google App engine.
Infrastructure as service (IaaS): provides virtual machines and other abstracted hardware and operating systems. Examples include Amazon EC2 and S3 [1], [2], Terremark Enterprise Cloud [6], and Rackspace Cloud [4].

The following deployment models are available for cloud computing services:

- *Private cloud*: services built according to cloud computing principles, but accessible only within a private network
- *Community cloud:* cloud services offered by a provider to a limited and well-defined number of parties
- *Public cloud:* available publicly - any organization may subscribe
- *Hybrid cloud:* a composition of two or more clouds (private, community or public)

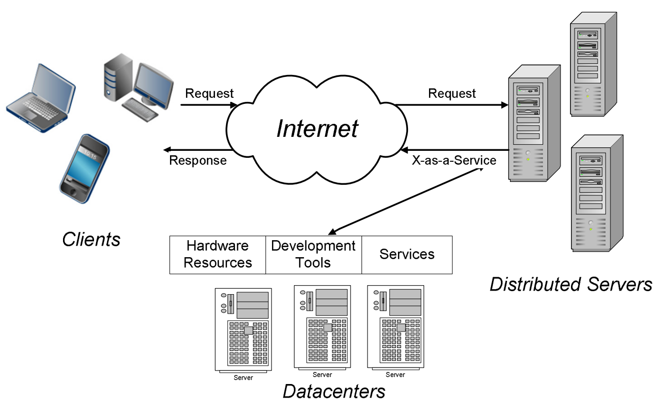


Fig. 1. Cloud Computing Topology

Thus, cloud computing provides a pool of highly scalable and easily accessible virtualized resources capable of hosting end-user applications exploited in a pay-as-you-go model. Cloud computing involves the following three basic components [7], which are illustrated in Figure 1: clients, datacenter and distributed servers. For many companies with highly variable IT needs, cloud computing can be an alternative to maintaining an expensive oversupply of in-house computing power. However, there are some major obstacles which hinder the adoption and growth of cloud computing. As every technological concept, cloud computing is not an exception in terms of trust and security issues. Once data are outsourced to a third-party cloud provider, several concerns arise about security, availability and reliability of data.

4 Branching Processes Theory

A branching process is a process where an initial random number of objects 'create' more objects of the same or different type, and these objects continue to 'create' other objects, with the system developing in accordance with some probability law. Branching processes are used to describe random systems such as population development, nuclear chain reactions and spread of epidemic disease. An example of such a process is a population of individuals developing from a single progenitor – the initial individual. It produces a random number of offspring, each of them in turn produces a random number of offspring; and so the process continues as long as there are live individuals in the population. Figure 2 is a graphic illustration of a general multilevel branching process. The branching process was proposed by Galton [9], and the probability of extinction was first obtained by Watson [8] by considering the probability generating function for the number of children in the nth generation.

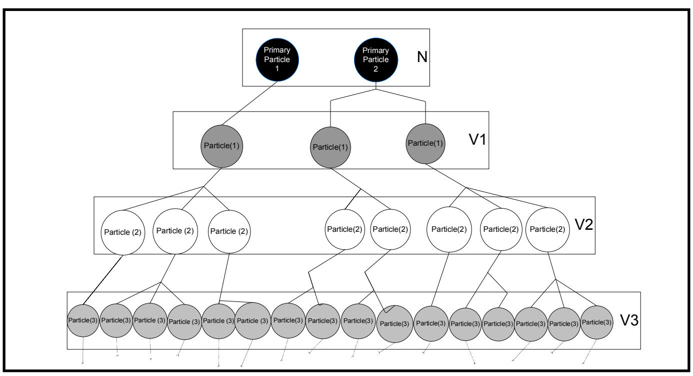


Fig. 2. Graphic illustration of a multilevel branching process

Let N be a random variable, with a probability distribution function $g_N(n)=P(N=n)$, with mean $\propto_N$ and variance σ_N^2. Let $\{X_i\}$ be a series of independent identically distributed random variables, with a common distribution $f_X(X)$ and with $\propto_X$ as the mean and σ_X^2 as the variance of each element in the series. The sum of N elements of the series $\{X_i\}$ is denoted by the following sum

$$V = X_1 + X_2 + \ldots X_N \tag{1}$$

The mean of V is denoted by

$$E[V] = E\left[X_i\right] \cdot E[N] \tag{2}$$

And the variance of V is denoted by

$$Var[V] = E[N] \cdot Var[X_i] + Var[N] \cdot E^2[X_i] \tag{3}$$

The distribution density function $f_V(v)$ of V can be derived from the basic formula for conditional probabilities

$$f_V(v = k) = P(V = k) = \sum_{n=0}^{\infty} P(N = n) P(X_1 + ... + X_n = k) \tag{4}$$

Let us denote $f_X(x)$ and $\phi_x(\omega)$ as the distribution density function and generation function of X_i respectively, and $g_N(n)=P(N=n)$ and $\phi_N(\omega)$ as the distribution density function and generation function of N respectively. For a fixed n, the distribution of the sum $X_1+...+X_n$ is expressed by the n-fold convolution of $\{f_X(x)\}$ with itself, due to the independence of the series $\{X_i\}$. Equation (4) can be written in a more compact form

$$f_V(v = k) = \sum_{n=0}^{\infty} g_N(n) \{f_X(x)\}^{n*} \tag{5}$$

This formula can be simplified by using the generating functions [15].

Branching processes theory can be applied for modeling in cloud computing environment. One of the most important issues in cloud computing environment concerns network efficiency and performance prediction. Where there are large quantities of data involved in an application, access to the data must be fast and reliable or the application's runtime will be excessive. From the viewpoint of a service provider, demands on the network are not entirely predictable. Branching processes theory helps us to model the dynamic demands for cloud services. More and more clients are informed on the service, one client from another based on random mechanism. This process is similar to epidemic of disease spread.

5 Data flow Prediction Model

To design effective and efficient network solutions for cloud environment and to understand and solve performance problems arising in communication networks providers require accurate models to describe network traffic. The main problem is to forecast the frequency of queries that are going to appear. To evaluate the performance of the proposed technique, we demonstrate how branching processes theory can be applied for building data flow prediction models.

The initial vertex of the graph is assigned $t=0$ (Figure 3), time when the original message reaches the host C_i. This point is taken as the reference time of receipt of queries to the cloud service provider's host center C_i. The initial vertex of a graph is based on the number of arcs equal to the quantity of primary needs. Arcs whose vertices correspond to the secondary queries come of the vertices of the graph corresponding to the initial query. We consider that the primary queries come into the provider's host randomly. The process of queries admission will be considered as a branching, while allowing that individual queries' paths are independent. The time intervals between $t=0$ and the time of requests admission are random variables with distribution function $F_1(x)$ and density $f_1(x)$. Obviously $f_1(x)$ represents the latency of the processes of information alerts. The number of initial requests coming to the provider for a certain period of time is the random variable

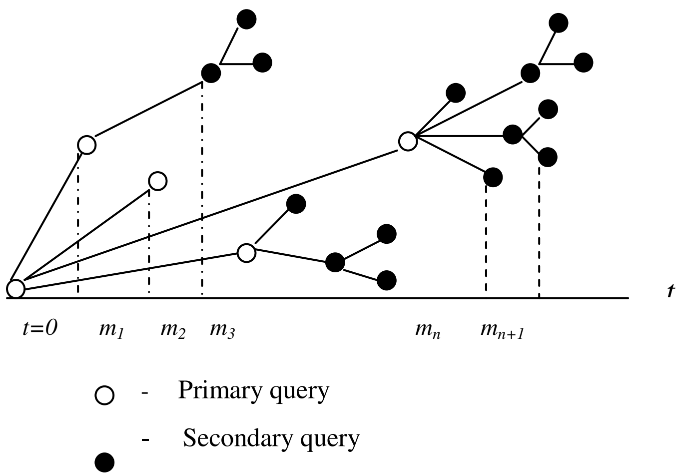


Fig. 3. A branching process of generating queries

$$\nu_1, P(\nu_1 = k) = p_{1k}, k = 0,1,2,...; \sum_{k=0}^{\infty} p_{1k} = 1 \tag{6}$$

We assume that the distribution of the primary query is binomial. The generating function corresponding to the binomial distribution of the primary query is:

$$G_1(u)=(q_0+q_1u)^n, \tag{7}$$

where u is the parameter of the generating function. The distribution function of the moments of primary queries receipt is exponential $F_1(x)=1-e^{-\lambda_1 x}$, where λ_1 is the volume of primary queries. After each initial query with probability p_0 does not appear any secondary query, and with probability $p_1=1-p_0$ there is at least one secondary request. We assume that the distribution of secondary queries generated by one primary request is subject to the binomial distribution with generating function

$$G_2(u)=(p_0+pu)^n , \tag{8}$$

where u is the parameter of the generating function. The distribution function of the time intervals between the moments of initial queries receipt and stimulated directly by them secondary queries is defined as $F_2(x)=1- e^{-\lambda_2 x}$, where λ_2 is the intensity of secondary queries receipt. The expectation of the number of requests in the time interval $[t, t+\tau]$:

$$M^*[t,\tau] = \begin{cases} \dfrac{nq_1\lambda_1\tau}{\lambda_1 - \lambda_2 p_0}\left\{(\lambda_1 - \lambda_2)e^{-\lambda_1 t} + \lambda_2 p_1 e^{-\lambda_2 p_0 t}\right\} \\ \qquad\qquad\qquad (\lambda_1 \neq \lambda_2 \mathrm{p}_0) \\ \dfrac{nq_1\lambda_1\tau}{p_0} e^{-\lambda_1 \cdot}(p_0 + p_1\lambda_1 t) \qquad \lambda_1 = \lambda_2 p_o \end{cases} \tag{9}$$

If $\dfrac{\lambda_1}{\lambda_2} < p_1$, then the distribution has a maximum. If $\dfrac{\lambda_1}{\lambda_2} \geq p_1$, then the distribution hasn't a maximum and decreases from the beginning. In case $\lambda_1 = \lambda_2 p_0$, for $p_0 \geq p_1$ distribution is monotonically decreasing; for $p_0 < p_1$ distribution has a maximum. Expectation function graphs of the number of queries are constructed for the following values: time units are $t=[0,130]$, the number of consumers which may create the primary requests is 10, the probability of the customer application with the primary request is $q_1=0.7$, the time interval of the request arrival $\tau=1$, the volume of the primary queries from the consumer at a time $\lambda_1=0.05$. The nature of the functional dependence is affected by primary and secondary queries intensity compliance. The Figure 4 contains plots of the expectation of the number of queries, in this case the probabilities of the primary p_0 and secondary p_1 queries: for $p_0 \geq p_1$ distribution is monotonically decreasing, for $p_0 < p_1$ the distribution has a maximum.

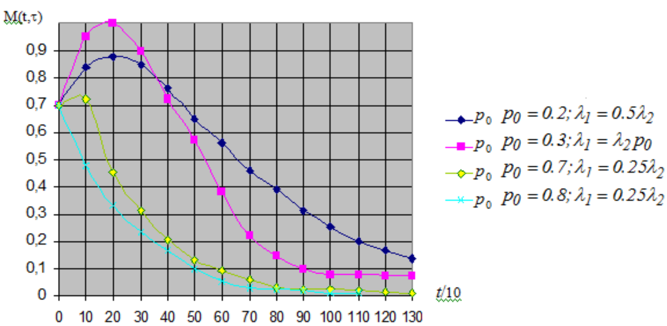


Fig. 4. The theoretical queries distribution functions over time

6 Research Methodology

Global giants like Amazon are definitely more cost-effective and less sensitive to daily, annual and other traffic imbalances than the local providers. In order to evaluate the performance of the proposed technique and to justify the further need for the model we are developing, the Russian IaaS market was analyzed. The table below presents the companies whose activity was examined. Sample organizations for analysis were selected based on the following criteria. Service content is similar to the services of a conventional hosting provider. Clients come from different industries. Service providers are with different backgrounds: they include new and established cloud suppliers (local).

Table 1. Russian IaaS (Hosting) Providers

#	*1*	*2*	*3*	*4*	*5*	*6*
Provider Name	Activecloud.ru	Clodo.ru	ISP Server	Scalaxy	Slidebar.ru	Selectel

In general, Russia's cloud hosting is not so much an alternative to Amazon Web Services, as a convenient substitute to traditional hosting. Most of the companies have encountered difficulties providing a comprehensive explanation of the dynamic characteristics of network traffic and had even experienced the bottlenecks that caused them to lose money. To measure how good the data flow prediction model is, experiments are carried out where cloud usage logs are analyzed seeing if our model can predict the loads seen in the logs.

7 Conclusion

Cloud computing is a model of delivering computing resources in which centrally administered computing capabilities are provided as services on-demand over the network to a variety of customers. As popularity of cloud services is growing rapidly, cloud-service providers must guarantee that data are processed effectively and transferred when and where they are needed. Unfortunately, it is extremely difficult to predict the exact performance characteristics and demands on the network at any particular time. In this paper we suggest that branching processes theory will fit for both describing the dynamics of cloud services demand and predicting the network traffic in cloud computing environment. We consider that more and more clients will know about cloud services, one client from another, like epidemic of disease spreads. The purpose of this paper is to provide a stochastic model that would be helpful for both consumers and providers of cloud-based services. On the one hand, traffic modeling helps to represent our understanding of dynamic demand for cloud services by stochastic processes, and on the other hand accurate traffic models are necessary for service providers to avoid "bottlenecks" and to improve quality of services provided.

References

1. Amazon Amazon elastic compute cloud, `http://aws.amazon.com/ec2/` (accessed March 2012)
2. Amazon: Amazon simple storage service, `http://aws.amazon.com/s3/` (accessed March 2012)
3. Microsoft: Windows Azure, `http://www.windowsazure.com/` (accessed March 2012)

4. Rackspace Cloud, http://www.rackspace.com/cloud/ (accessed March 2012)
5. Salesforce CRM, http://www.salesforce.com/eu/ (accessed March 2012)
6. Terremark Enterprise Cloud, http://www.terremark.com/services/infrastructure-cloud-services/enterprise-cloud.aspx (accessed March 2012)
7. Tsaravas, C., Themistocleous, M.: Cloud Computind & E-Government Myth or Reality? In: tGov Workshop 2011 (tGOV 2011), March 17-18 (2011)
8. Watson, H.W.: Solution to problem 4001: Educational Times, August 1 (1873)
9. Galton, F.: Problem 4001: Educational Times, April 1, p. 17 (1873)
10. Mohammed, A.M., Agamy, A.F.: A Survey on the Common Network Traffic Sources Models. International Journal of Computer Networks 3(2) (2011)
11. Frost, V.S., Melamed, B.: Traffic Modeling for Telecommunications Networks. IEEE Communications (March 1994)
12. Ferrariand, D., Verma, D.: A Scheme for Real Time Channel Establishment in Wide Area Networks. IEEE Journal on Selected Areas in Communications 8(3) (April 1990)
13. Erramilli, Sing, R.P., Pruthi, P.: Chaotic Maps as Models of Packet Traffic: The Fundamental Role of Teletraffic in the Evolution of Telecommunications Networks. In: Proc. of the 14th ITC, June 6-10 (1994)
14. Ribeiro, V.J., Riedi, R.H., Crouse, M.S., Baraniuk, R.G.: Simulation of non-Gausian Long-Rang-Dependent Traffic Using Wavelets. In: Proceeding SIGMETRICS 1999, April 9 (1999)
15. Cohen, I., Golan, R., Rotman, S.: Applying Branching Processes Theory for Building a Statistical Model for Scanning Electron Microscope Signal. Optical Engineering 39(01)
16. Harris, T.E.: The theory of branching processes. Springer, Berlin (1963)

OpenCASE– A Tool for Ontology-Centred Conceptual Modelling

Robert Pergl and Jakub Tůma

Department of Information Engineering,
Faculty of Economics and Management,
Czech University of Life Sciences, Prague, Czech Republic
{pergl,jtuma}@pef.czu.cz

Abstract. OpenCASE, an original CASE tool supporting conceptual modelling is presented in this paper. The CASE tool has been developed during the research focused on the ontology-centred conceptual modelling. It provides a strong emphasis on terms and their relations while supporting standard notations (now BORM, other notations are planned). The tool has an open plug-in-based architecture founded on the Eclipse platform, which makes the tool modular and extensible. The knowledge base of the models may be accessed via an API and thus used to implement verifications, various calculations (statistics), to transform models to outputs (reports) and to make inner transformations (e.g. normalisation). The architecture of the tool is briefly mentioned as well.

Keywords: CASE Tool, Eclipse platform, conceptual modelling, ontological analysis, BORM method.

1 Introduction

This contribution addresses the discussion of the importance of diligent ontological analysis during the enterprise IS modelling presented in [13], where the author explains the importance of ensuring the *consistency* between various models (and inside each model) and concludes (besides others) the need of a "quality CASE tool support".

In this paper, we would like to present our advancements in designing and implementing a CASE tool to support ontology-centred modelling: OpenCASE. OpenCASE [12] is a CASE tool designed to support the research in the field of conceptual modelling and ontologies. It is built upon the Eclipse framework [8] and it utilizes many of its advanced possibilities (see section 4). Right now, we have implemented the BORM method's Business Architecture Diagrams and Object Relation Diagrams ([7],[1], [10]) as a proof-of-concept of ontology-centred modelling and OpenCASE's philosophy and design[1].

[1] Apologies for the readers: due to lack of space in this short paper, we do not provide a BORM introduction here.

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 511–518, 2012.

2 Goal and Methodology

The goal of the contribution is to present `OpenCASE`, an original ontology-centred CASE tool. We present the philosophy behind the tool, its practical utilisation and its architecture.

3 Ontology-Centred Modelling

3.1 Entities vs. Elements

To deal with ontology generally means to deal with *terms* and their *relations*[2]. Conceptual modelling using the ontology-centred approach thus needs to fully support *tracking of distinct terms and their relations* throughout the models. This practically means a transition from *visually-centred* to the *ontology-centred* CASE tools architecture. We may find notions of this approach in some CASE and Meta-CASE tool like Craft.CASE [3], MetaEdit+ [11] and others, however we were focused on total concept purity in separating the Domain Layer and the Domain Model Layer while maintaining the relation between the elements (Figure 1[3]).

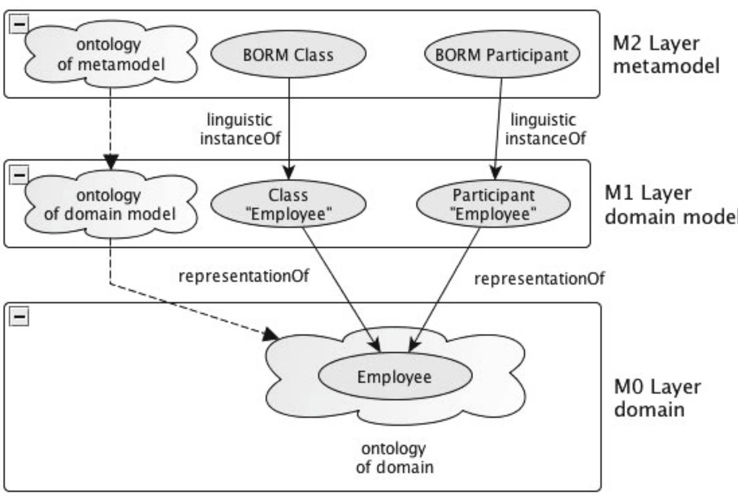


Fig. 1. Layered architecture of ontologies (taken from [13])

We implemented the concept of *ontologically equivalent elements* introduced in[13]:

[2] A more thorough introduction to ontologies and their relation to conceptual modelling and the BORM methodology may be found in our original paper [13].

[3] The layered architecture of ontologies has already been published in its rudimentary form numerus times over the past 30 years, see e.g. [4].

Definition 1. *We say that the model element x is* ***ontologically equivalent*** *to the model element y if and only if there exist relations $representationOf(x, t)$ and $representationOf(y, t)$, where t is an element of the CMoD*[4].

We implemented this concept by strictly discerning the domain *entities* and model's *elements*. An "entity" represents a domain object in the M0 layer in Figure 1, while the "element" is an object in the M1 layer, being a particular instance of the M2 layer element. Practically, let us suppose we deal with a `Customer` entity being modelled by a Participant `Customer` graphically represented according to the BORM methodology as a rectangle with a solid border and pale blue filling, which is an instance of the `Participant` concept.

There is a relation 1:N between entities and elements: each entity (layer M0) may be represented by several elements (layer M1): a Customer may play its role in several diagrams, thus being represented by a visual element in each, while being the same entity's $representationOf$. There is a screenshot in Figure 2 showing the `OpenCASE`'s support for tracking entities with respect to elements. The left panel shows all the entities. If we unfold an entity, we see all its elements. In Figure 2 there are 3 participant elements that represent entity `Customer`[5]. When we click an element, its full path is revealed in the status line. Double-clicking an element takes us to the appropriate diagram and selects the element.

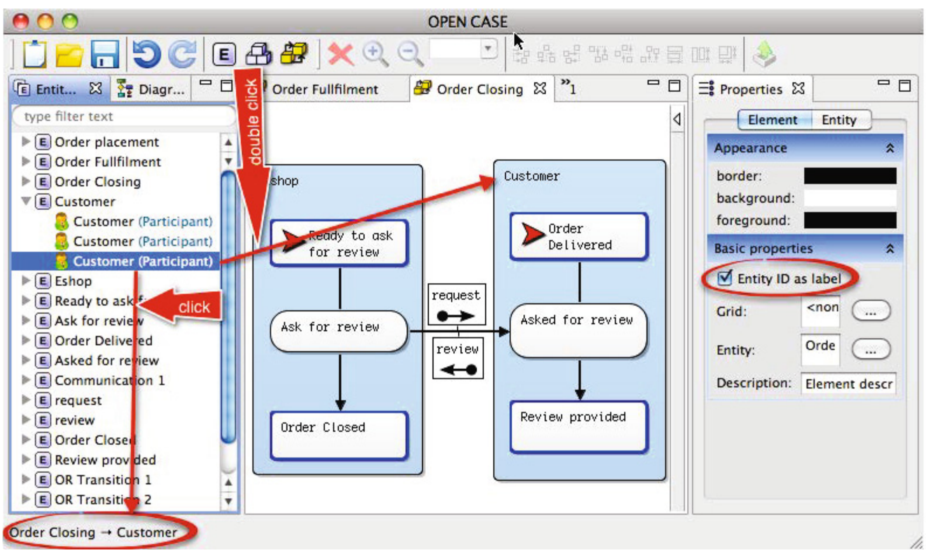


Fig. 2. Entities and Elements in `OpenCASE`

[4] CMoD = Concept Map of Domain . . . a graph of domain terms and their relations.

[5] Generally, there may be various element types representing one entity, e.g. there may be as well a data class `Customer` that would describe customer's attributes.

Tracking the entities is a concept that enables us to:

Ensure elements' consistency – Elements usually take the name of their entity, so renaming an entity automatically renames all the attached elements. Thus if we have an element `Customer` participating in several diagrams and we change its entity's name to `Client`, all the attached elements get automatically renamed. In case we do not want an element to automatically take its entity's name – this may be the situation in multilingual diagrams or languages that use inflection – we may disable the implication "entity name → element name" (option `Entity ID as label`, red ellipse on the right in Figure 2). We need to rename the element by hand, however with the comfort of having the list of all elements that are *representationOf* the entity.

Facilitate impact analysis – When some change occurs to a domain entity, we may easily track the impact to the model, i.e. the elements in the model that may need attention due to the change.

Use the model knowledge base – Elements represent some information we have about the entity – its roles and relations in the domain. We may design and run reports, statistics, optimizations and reasoning based on this knowledge. `OpenCASE` provides a full API to this model knowledge base – see subsection 3.4.

3.2 Business Properties

Another concept of ontology-centred modelling implemented in `OpenCASE` are *business properties*. Inspired by the success of this concept in the Craft.CASE tool [3], we implemented a sort of meta-modelling layer enabling to specify custom domain (business) properties. Compared to Craft.CASE we did not limit business properties just to classes of elements (Participant, State, Activity, ...), but we made possible to attach a business property to an element (thus having just one instance) or to an element class (thus having several instances) – Figure 3. Because diagrams are elements as well, they may have business properties orthogonally assigned, too (the author, version, etc.). If we look at states, activities, diagrams, participants and other elements as UML classes, business element class BPs would correspond to class attributes, while element properties would correspond to instance attributes.

3.3 Internal Knowledge Base

The tool is database-centred, i.e. it maintains an internal knowledge base containing functions, scenarios, diagrams, entities, elements together with their graphical properties. Internally, they form nodes of a graph structure and their relations are represented as edges, thus various graph algorithms may be applied on the knowledge base [17]. Graph traversals and graph transformations are probably the most useful and may implement operations like

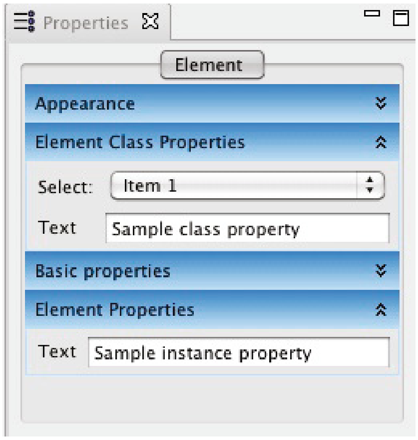


Fig. 3. Element Properties and Element Class Properties in `OpenCASE`

- *Listings*, like all input/output flows from/to a participant.
- *Calculation of metrics* (like numbers of states and activities in participants) that may be used for *complexity estimations* [16], [15].
- *Calculation of statistics*, e.g. about dataflows and communications (which participants communicate the most/least, above/below average, etc.).
- *Semantics checks*: there is a starting state in every participant, at least one final state[6], ...
- *Conceptual normalisations* [9].
- Any further custom reporting / calculations / processing.

3.4 Model API

We see diagrams as a convenient way how to specify the model and visualize the model to a business user, however the true power lies in its underlying knowledge base. We designed `OpenCASE` to transparently reveal its API (Application Programming Interface) of the model's structure. Using this API, a programmer may iterate through the model's elements and entities, make verifications, perform various calculations (statistics), transform them to some sort of output (reports) and make inner transformations (e.g. normalisation).

The API is self-documented in the form of UML Class diagrams thanks to the Ecore framework (see section 4). An example of the OR diagram metamodel is in Figure 4.

4 `OpenCASE` Implementation

`OpenCASE` is implemented entirely in the Java programming language utilizing the Eclipse Platform and various modelling frameworks from the Eclipse Mod-

[6] According to the BORM method, there may be exceptions to these rules, see [7] for more details.

eling Project (EMP). The Eclipse Rich Client Platform (RCP) offers a very powerful foundation since it provides an extensible component system and a platform for creating complex applications with rich user interfaces.

Due to lack of space in this short paper, we do not provide an overview of the RCP platform. The reader may read about it on the Internet or in the literature – we highly recommend [2], [5], [8], [14].

The core of the OpenCASE project is the OpenCASE written as an RCP application. There are 10 essential plug-ins constituting the core of the application called OpenCASE Workbench. The workbench is just the user interface without any diagram editing capabilities. Diagram manipulation is performed by the remaining plug-ins.

Each feature has a core plug-in having an ID with no suffix, e.g., `org.opencase.diagrams`. Such a plug-in is almost entirely generated from an Ecore model and it implements the basic behaviour of the modelled domain. An example of Ecore model is in Figure 4.

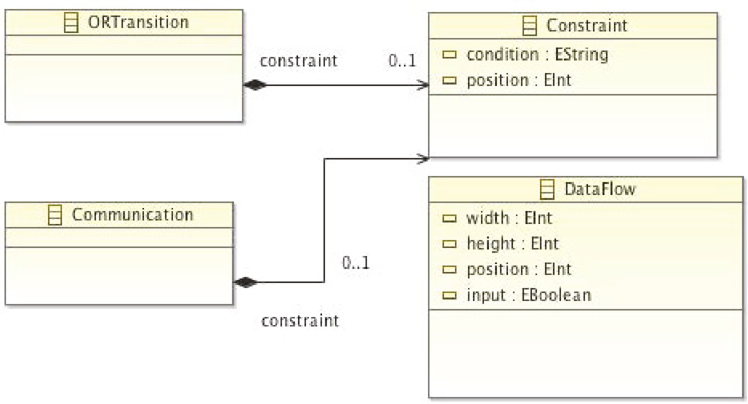


Fig. 4. A part of Ecore model of BORM's Object Relationship Diagram

Currently, there are several plug-ins being developed, compiled and deployed in separation from the core of OpenCASE. Thanks to Eclipse plug-in system such components can be installed right into the running OpenCASE from an archive or internet update site.

5 Summary, Conclusions and Future Work

OpenCASE is an attempt to bring the ontology-centred modelling into everyday life and profit from research achievements while at the same time to provide an open platform for further research. That provided a challenge to implement theoretical results into suitable software implementation and to build a user-friendly tool with features like keyboard shortcuts, complete undo, aligning and distribution of graphical elements, batch operations, etc.

We put a high focus to implement the whole ontology chain, i.e.

1. **Input** – *How to input the terms and their relations in synergy with a concrete notation.* We addressed this issue by separating the identity (entity) from its representation (element) – subsection 3.1
2. **Processing** – *How to access the ontology and manipulate it by transformations and various algorithms* (verifications, normalizations, optimizations, etc.). We built an application programming interface (API) to access the model's knowledge base. The API architecture is documented by UML (Ecore) diagrams[7] – subsection 3.4.
3. **Output** – *How to export the ontological knowledge contained in the model.* Exporters plug-ins handle this task. Exporter plug-ins are implemented as Eclipse plug-ins and they may be implemented to perform an export to various human-readable formats (TXT, HTML, LaTeX, ODT, PDF, ...) or formats suitable for machine processing (CSV, XML, JSON, OWL, ...), or it may perform the export directly into relational database or reveal the knowledge base as a service (SOAP, REST).

At the time of writing this contribution, the modelling core is completely implemented, being further fine-tuned and improved. As for the plug-ins, several output plug-ins are developed (TXT, HTML, LaTeX). We are also working on implementing models simulations and support for optimizations. A huge step toward the holistic ontology-centred conceptual modelling will be implementing other types of diagrams, especially data-structure diagrams (UML Class Diagrams and OntoUML, [6]) and providing a means to make ontologic relations to the process diagrams.

Acknowledgements. This contribution was elaborated with a support of grant no. 20121059 of Grant Agency of The Faculty of Economics and Management of the Czech University of Life Sciences in Prague.

References

1. Brožek, J., Merunka, V., Merunková, I.: Organization Modeling and Simulation Using BORM Approach. In: Barjis, J. (ed.) EOMAS 2010. LNBIP, vol. 63, pp. 27–40. Springer, Heidelberg (2010)
2. Clayberg, E., Rubel, D.: Eclipse Plug-ins. Addison-Wesley Professional (2008)
3. Craft.CASE Tool, http://www.craftcase.com
4. Gasevic, D., Djuric, D., Devedzic, V.: Model Driven Engineering and Ontology Development, 2nd edn. Springer (2009)
5. Gronback, R.C.: Eclipse Modeling Project: A Domain-Specific Language (DSL) Toolkit. Addison-Wesley Professional (2009)

[7] Actually, as was explained, the situation is opposite: the internal structures are *generated* from the Ecore diagrams, which makes a powerful mechanism to maintain specification-implementation consistency. Nevertheless, this is irrelevant for the API's user.

6. Guizzardi, G.: Ontological Foundations for Structural Conceptual Models. Telematica Instituut Fundamental Research Series No. 15 (2005)
7. Knott, R.P., Merunka, V., Polák, J.: The BORM methodology: a third-generation fully object-oriented methodology. Knowledge-Based Systems 16(2), 77–89 (2003)
8. McAffer, J., Lemieux, J.-M., Aniszczyk, C.: Eclipse Rich Client Platform: Designing, Coding, and Packaging Java Applications. Addison-Wesley Professional (2005)
9. Molhanec, M.: Some Reasoning Behind Conceptual Normalisation. In: Information Systems Development, pp. 517–525. Springer Science+Business Media, Berlin (2011)
10. Molhanec, M., Merunka, V.: BORM: Agile Modelling for Business Intelligence. In: Business Intelligence and Agile Methodologies for Knowledge-Based Organizations: Cross-Disciplinary Applications, pp. 120–130. IGI Global, Hershey (2011)
11. MetaEdit+, `http://www.metacase.com/cases/borm.html`
12. Tool, `http://www.opencase.net` (in construction)
13. Pergl, R.: Supporting Enterprise IS Modelling Using Ontological Analysis. In: Barjis, J., Eldabi, T., Gupta, A. (eds.) EOMAS 2011. LNBIP, vol. 88, pp. 130–144. Springer, Heidelberg (2011)
14. Steinberg, D., Budinsky, F., Paternostro, M.: EMF: Eclipse Modeling Framework. Addison-Wesley Professional (2008)
15. Struska, Z., Merunka, V.: BORM points – New concept proposal of complexity estimation method. In: ICEIS 2007: Proceedings of the Ninth International Conference on Enterprise Information Systems: Information Systems Analysis and Specification, 9th International Conference on Enterprise Information Systems (ICEIS 2007), Funchal, Portugal, June 12-16, pp. 580–586 (2007)
16. Struska, Z., Pergl, R.: BORM-points: Introduction and Results of Practical Testing. In: Filipe, J., Cordeiro, J. (eds.) ICEIS 2009. LNBIP, vol. 24, pp. 590–599. Springer, Heidelberg (2009)
17. Valiente, G.: Algorithms on Trees and Graphs. Springer (2010)

Business Service Integration Using Pattern Composition

Jeewanie Jayasinghe Arachchige and Hans Weigand

Tilburg University, P.O.Box 90153,
5000 LE Tilburg, The Netherlands
{J.JayasingheArachchig,H.Weigand}@uvt.nl

Abstract. Smooth integration of services is one of the key benefits of service-oriented enterprises. But the key questions are "Does the selected service address the real business need?" and "How to select the right service?". Separating the business choices and technical choices is necessary in order to structure the solution to these questions. In this research, the two perspectives are handled by embedding service thinking at business level design and secondly, by supporting the integration with available services and process templates. To reach this combined goal, we present a pattern-based approach for business service integration.

Keywords: Business Service Pattern, Pattern Composition, BSRM, Business Service Integration.

1 Introduction

Nowadays organizations are rapidly adopting SOA into their information systems and the business strategy, because of its emerging advantages. This environment creates an opportunity to such organizations to interact dynamically as service consumers and service providers making use of a service marketplace to design, offer and consume services [1]. On the other hand, it is claimed [2] that "the full potential of SOA is only realized when it is applied as an architecture for business design". Hence, service oriented business thinking is important when integrating the software services. Therefore the objective of this research work is, first of all, how to embed service thinking at business level and, secondly, to support the integration of available services and process templates. To reach this combined goal, we present a pattern-based approach for business service integration.

There exists some relevant work on pattern-based service integration. The research in [1] presented a pattern-based modeling approach to achieve the unforeseen integration of services into extensible enterprise systems. This framework demonstrates service integration to the presentation (HCI) layer using adaptation patterns that group common patterns of model elements and their relationships. Even though this approach enables an integrator to model or design the relevant integration aspects on a higher abstraction level than implementation-level, service thinking at the business level is not addressed. The work is related to the concept of plug-in technologies that allow the development and installation of web 2.0 applications.

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 519–530, 2012.

Based on the example of SAP's Enterprise Services, authors of [3] describe a representational model that integrates both service and data by consolidating existing models and patterns used during the service design process. On top of this model, they created a metadata repository based on a list of ES and their respective metadata. Both representational model and metadata repository represent the basis of the iterative search of software services. Even though the business objects are one of the ingredients of the representational model, there is no clear basis of selecting business objects to their model and their focus is on the presentation layer of the ES by providing a pathway to service matchmaking.

There is also relevant research that incorporates business thinking to the service design and aims at mapping those designs to the software level. One good approach is value based service design. [4,5,6,7] present an MDA approach to design and transform services from CIM, PIM to PSM. However, the focus of all these approaches is on service design and transformation, not on service integration. Some other researches can be found in the literature for example [1] and [3], with a focus on service integration. One problem here is that business choices and technical choices are all dealt with together at a technical layer and have an exclusive software engineering perspective.

The original contribution of this paper is that we extend the intuitive business service patterns presented in [8] to facilitate modeling the business services at a conceptual business level, by proposing a model-driven method of using these patterns in service integration. Business service patterns are represented using our previous work [4], BSRM (business service and resource modeling) language, which is based on the REA business ontology [9].

This paper is structured as follows: Section 2 provides a background which describes the BSRM ontology – the basis of developing patterns. Section 3, consists of four parts. Part one and two describe the generic business service patterns for service outsourcing and the sub-services. Part three describes the pattern composition using operators and section three ends with providing design steps. Service integration is explained in the section 4. The conclusion is given in the section 5.

2 Background

Using an MDA approach, we have introduced a new business service and resource modeling language - BSRM based on the Resource-Event-Agent (REA) business ontology in our previous work [4]. BSRM is capable to design the business activities in a company with service perspective at CIM level using simple modeling notation. The constructs of the BSRM language and their relationships are grounded in a meta-model (Fig. 1) which provides comprehensive specification using UML notation. We distinguish two service specializations: exchange service and conversion service, corresponding to the two basic REA dualities. Each of them corresponds to a group of decrement and increment economic events in REA.

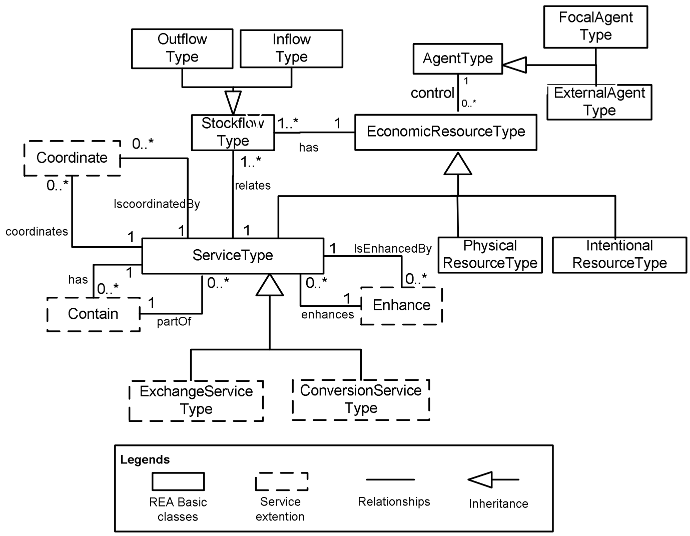


Fig. 1. BSRM meta-model

Based on the service classification model [6], we identified two different service roles. The concept of *enhancing* service which adds value to the any other service is introduced as one of them. Considering the situation where core-service realization involves multiple value activities and it makes sense to view these value activities as independent services that are shared by different contexts, we identified the next category of services as *sub-services*. A distinction is made between *core sub-services* and *coordination services*. The former are used in the realization of the composite service by manipulating physical resources and the later are used in the realization of the composite service, and coordination of the core sub-services, by manipulating intentional resources. The relationship between economic resource type and service type is defined as REA stockflow, further specialized as inflow and outflow. Following REA, we include the agent concept to the meta-model. An agent is an individual or organization capable of having control over economic resources. Agent type is classified into two, namely focal and external. The focal agent type is the individual or the central organization who intends to view its business with service perspective. External agent type represents the outside agents who are involved in the service value chain. The relationship between the economic resources and agent is "control". Note that agents are not explicitly represented in the BSRM model. A BSRM model always takes the perspective of a particular agent. The design decision to deemphasize the agents/owners is in line with the SD-logic approach in which co-creation is more important than ownership.

We used a simple modeling notation for BSRM. A summary of the modeling notation is given in the table 1.

Table 1. Summary of the modeling notation

Services are denoted as rounded rectangle (Conversion services are colored/filled rounded rectangles and exchange services are not filled with a color.)	
Physical resources are denoted as rectangles	
Intentional resource are denoted as dashed rectangles	
Part of Relationship Co-subservices are denoted as Coordination relationship is denoted as	 +
Enhancing relationship is denoted as	+
Stockflow relationship Inflow Outflow	 ← →

3 Business Service Patterns (BSP) and Composition

The purpose of using business service patterns is twofold. Firstly, the reuse of patterns reduces the designing time. Secondly, it assures that the designer does not violate the domain concepts. We introduced generic business service patterns for exchange and conversion processes in [8]. In this section, we present just two generic service patterns - service outsourcing and the service which has multiple sub-services. These patterns are modeled with BSRM notation and from the perspective of the focal agent.

3.1 Outsourcing a Service

Description:

Companies are outsourcing services to another party due to several reasons. Some of them are lack of own resources, lack of competencies and expertise knowledge and when the outsourcing is cost effective.

Pattern:

The business service pattern for service outsourcing is shown in Fig. 2. Outsourcing is based on the exchange pattern. The company has to pay a fee for the external agent to get their service. Hence there exist *give* and *take* relationships between *ServiceOutsourcingExchange* and the *money* and the service, respectively. The service which is outsourced has stock inflow and outflow relationships with resources. In particular, it *uses* resources (resource 1) and it *produces* another resource (resource 2) - the meaning of "produce" can be actual production or adding value.

Parameters:

s: ServiceType (the outsources service), FA, EA: agent (focal agent, external agent)

Constraints:

- At least one resource outflow of the service belongs to the focal agent. This ensures that the service has value for the focal agent.
  $\exists$r: ResourceType [produce(s,r) $\wedge$ control(FA,r)]
- At least one resource inflow belongs to the external agent. This ensures that the external agent really contributes something.
  $\exists$r: ResourceType [use(s,r) $\wedge$ control(EA,r)]

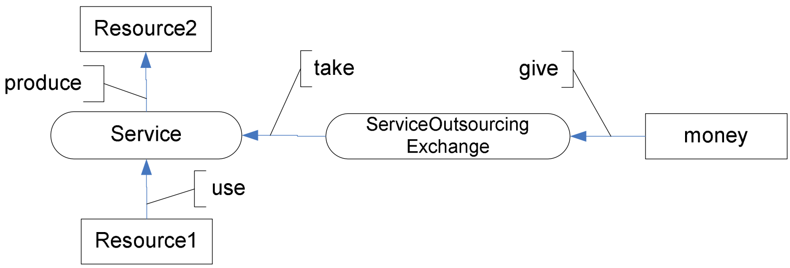


Fig. 2. Service Outsourcing pattern

3.2 Sub-service Pattern

Description:

Sub services are coming into play when the core service has to be realized by multiple activities. For example *produce* is a core service and it has *assemble*, and *inspection* sub-activities. When decomposing the composite service, it is important to analyze the relationship between resources with the sub-activities and their constraints.

Pattern:

The business service pattern for sub-service is shown in Fig. 3. Sub-services have a part-of relationship with the composite service.

Parameters:

s: ServiceType (the composite service), *s1* and *s2* are sub services

Constraints:

- At least two sub services are to be defined.
  $\exists$s1,s2: ServiceType [part-of(s1,s) $\wedge$ part-of(s2,s) $\wedge$ s1$\neq$s2]
  Note:
  Number of sub-services can be more than 2. Following the pattern expansion method in [10], number of sub-services can be increased. In the pattern expansion, the model has a fixed part and a variable part; we consider the sub services and its related resources as the variable part.
- Intermediate outflow resources have to be consumed by another sub-service.
  $\forall$r: ResourceType [[$\exists$s1: part-of(s1,s) $\wedge$ outflow(s1,r)] $\Rightarrow$ outflow(s,r) $\vee$ [$\exists$s2: part-of(s2,s) $\wedge$ inflow(s2,r)]]

- Intermediate inflow resources have to be produced by another sub-service.
  ∀r: ResourceType [[∃s1: part-of(s1,s) ∧ inflow(s1,r)] ⇒ inflow(s,r) ∨ [∃s2: part-of(s2,s) ∧ outflow(s2,r)]]
- Inflow resources must be consumed by a sub-service, outflow services must be produced by a sub-service.

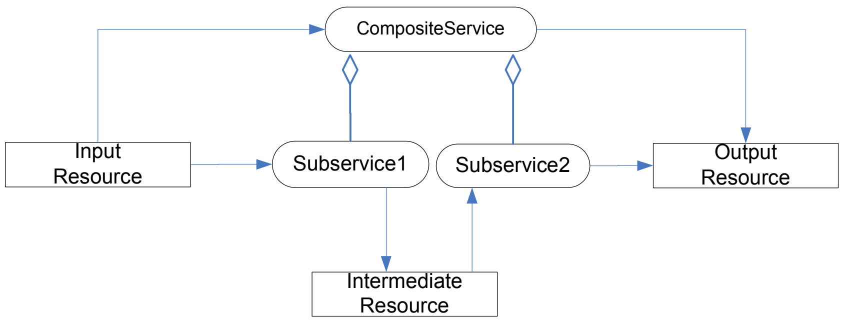


Fig. 3. Sub-service pattern

3.3 Pattern Composition with Business Pattern Operators (BPO)

The enterprise model is a composition of BSPs that always starts from a generic enterprise model (for trading company, for manufacturing etc). BSPs can be combined using business pattern operators each time the enterprise model is expanded. This section describes the operators used in business service pattern composition. A pattern can be viewed as a graph consisting of nodes and edges. In this section we present several operators to merge and expand patterns. The term "pattern" is used for both the basic patterns given in the library and the compositions of patterns. We follow the definitions of [10] for pattern composition, annotating and expansion which originate from the category theory.

1. Merge Operator

The merge operator enables to combine two patterns and compose a new pattern. We follow the definition b-4 in [10] in which the composition is described using the technique "pushout" by gluing the two objects along a common sub object. The following acronyms are used for easy representation of the merge operation.
BSP1, *BSP2*: business pattern 1 and 2 respectively;
K: common object to BSP1 and BSP2;
Morphisms m1, m2 to BSP1 and BSP2: BSP1 $\overset{m1}{\leftarrow} K \overset{m2}{\rightarrow}$ BSP2 respectively.
The possible candidates for the common object can be a *physical resource*, an *intentional resource,* a *conversion/ exchange service,* a *coordination service* or an *enhancing service*.

The following example (Fig.4) illustrates the merge operation. Consider the situation where we need to outsource the delivery service when self delivery is not possible.

Pattern (a) in Fig.4 represents the business service pattern of delivery service. Delivery is a service used in the sales and it adds value to the product. It uses forklift and truck 1 as physical resources. Pattern (b) in Fig.4 shows the delivery outsourcing pattern. It is an exchange service which takes the delivery service and gives money. Delivery service in the outsourcing pattern uses truck 2. Merging product delivery with delivery outsourcing is done through the common object “delivery service”. We build the constraints of the new model as a union of constraints which relate to the common object.

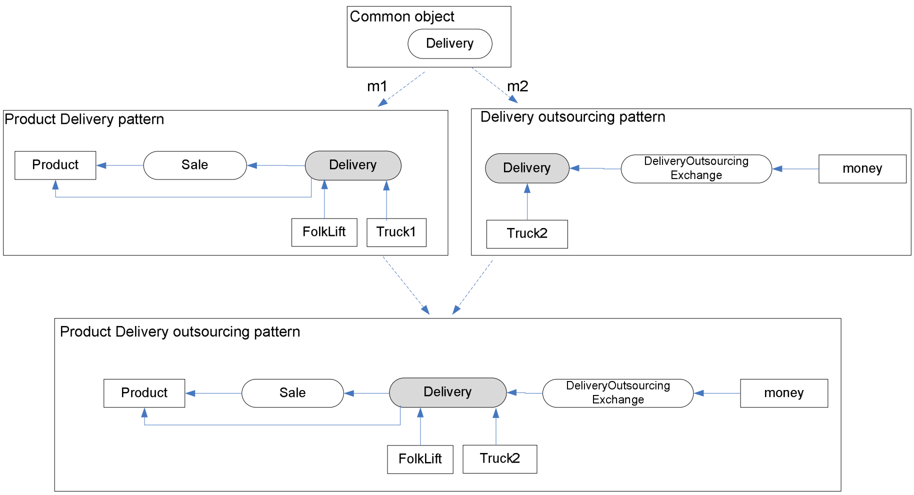


Fig. 4. Example for merge operator

Constraint of delivery service s in product delivery pattern (in fact, this constraint holds for *any* service type):

$\exists$r1,r2: ResourceType [inflow(s,r1) $\wedge$ outflow(s,r2) $\wedge$ r1$\neq$r2]

Constraint of delivery service s in delivery outsourcing pattern (cf. 3.2):

$\exists$r: ResourceType [produce(s,r) $\wedge$ control(FA,r)]

$\exists$r: ResourceType [use(s,r) $\wedge$ control(EA,r)]

These constraints are not inconsistent. The union can be written as:

$\exists$r1,r2: ResourceType [inflow(s,r1) $\wedge$ control(FA,r) $\wedge$ outflow(s,r2) $\wedge$ control(EA,r) $\wedge$ r1$\neq$r2]

In the result, the delivery service has one inflow resource from the external agent (truck 2) and one outflow resource of the focal agent (product).

2. Decompose operator

This operator is used when the service is realized by multiple activities (sub-services) that have to be distinguished, for instance, because they involve different actors and responsibilities, or because it allows for more flexible sourcing of the sub-services. Decomposition has to be defined in three steps.

Step 1: Pattern Expansion (if there are more than 2 sub-services)
In the first step, the sub- service pattern has to be expanded if there are more than 2 sub-service. Pattern is expanded as described in section 3.2 which is based on the pattern expansion definition [10].

Step 2: Derive the domain specific model
The second step is deriving the domain specific model by pattern annotation, again based on the definition in [10]. They use pattern annotation to represent the model with specialize vocabulary in a specific domain. Using a triple graph which consists of the source graph -domain specific pattern, target graph- vocabulary pattern and the correspondence graph- source and the target relates through morphisms, the specific model is derived. Here we use an example of bike producing which has assemble and inspection sub activities. According the constraints in sub-service pattern we do not need expansion to the generic sub- service structure because it has 2 sub- services. By adapting vocabulary pattern which consists of the names of specific domain- bike producing, we derive the decompose domain specific model for bike producing (Fig.5(b)).

Step 3: Merge the composite pattern with decomposed pattern
The third step is merging the decomposed pattern with the composite pattern as described in 'merge operator' section. Fig. 5 shows the decomposition. Producing a bicycle is a composite service and it can be realized through assemble and inspection sub services.

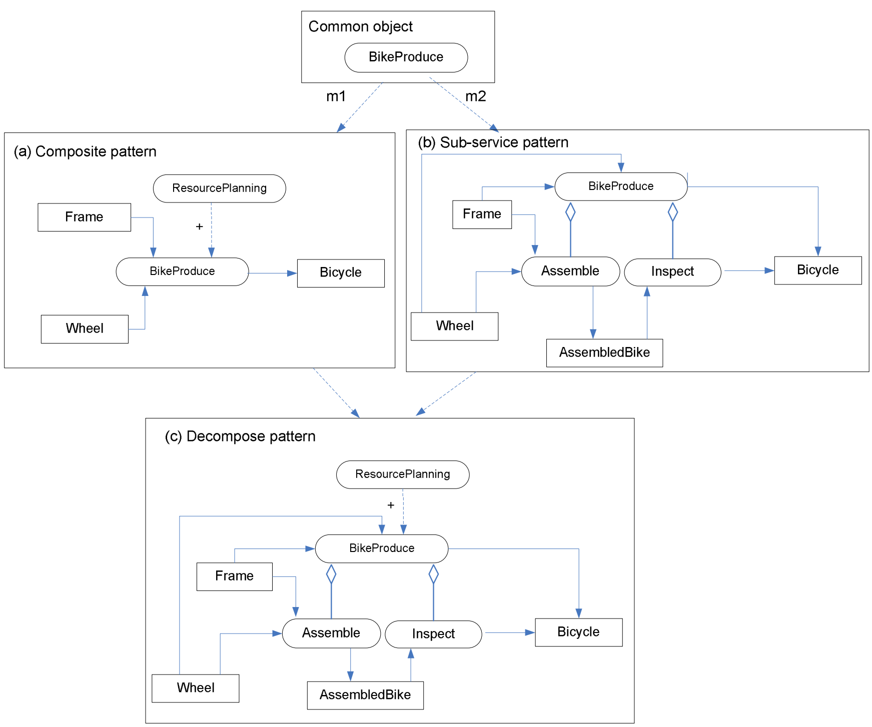


Fig. 5. Decomposition

Constraints for composite service: (s: ServiceType)

$\exists$r1,r2: ResourceType [inflow(s,r1) $\wedge$ outflow(s,r2) $\wedge$ r1$\neq$r2]

Constraints for sub-service pattern:

Inflow resources must be consumed by a sub-service, outflow services must be produced by a sub-service

$\exists$r1,r2: ResourceType [inflow(s,r1) $\wedge$ outflow(s,r2) $\wedge$ r1$\neq$r2] $\wedge$[inflow(s1,r1) $\wedge$ outflow(s2,r2) $\wedge$ r1$\neq$r2]

All the other constraints related the sub-service pattern remain unchanged in the decomposition model.

The union of above two constraint can be written as:

$\exists$r1,r2: ResourceType [inflow(s,r1) $\wedge$ outflow(s,r2) $\wedge$ r1$\neq$r2] $\wedge$[inflow(s1,r1) $\wedge$ outflow(s2,r2) $\wedge$ r1$\neq$r2]

3. Specialization operator

The specialize operator is used when a service is further qualified with a specific domain. It is also achieved in two steps. First the pattern has to be expanded with the number of specializations (if there is more than one). Then, using pattern annotation, the domain specific model has to be derived. For example, *bike produce* is a specialization of the *product produce* service.

Constraint for specialization of a service

- At least one specialization is to be defined.

  $\exists$s1: ServiceType [specialize (s1,s) $\wedge$ s$\neq$s1]

Constraint for specialization of resources

$\forall$r: ResourceType[inflow(r,s)] $\Rightarrow$

$\exists$ r1: ResourceType inflow(r1,s1) $\wedge$ subtype(r1,r2)

$\forall$r: ResourceType[outflow(r,s)] $\Rightarrow$

$\exists$r2: ResourceType outflow(r2,s1) $\wedge$ subtype(r1,r2)

3.4 Design Steps for Enterprise Information Systems

Given the service patterns and pattern operators, a service design method can be developed. The following activities are intended to describe the design steps to achieve such model. All the steps after the first step are not strictly ordered.

Step 1: $\phi \rightarrow$ enterprise model

Step 2: decomposition: $\{s_0\} \rightarrow \{s_0, s_1, \ldots s_n\}$, where part-of($s_i$,s_0)

Step 3: coordination: $\{s_{1,\ldots,} s_n\} \rightarrow \{ s_0 ,s_{1,\ldots,} s_n, c_i,\}$, where part-of($c_{i,,} s_0$), coordinate($c_{i,,} s_0$)

Step 4: extension: $\{s_{1,} s_2\} \rightarrow$ merge($s_{1,} s_2$)

Step 5: enhancing: $\{s\} \rightarrow \{s,e\}$, where enhance(s,e)

Step 6: specialization

Step 7: outsource, reengineer

The design starts top-down with the generic enterprise model, to ensure the completeness. In the second step, identify specific services by decomposing the enterprise model. This is done using domain models. Once multiple sub- services exist, they have to be coordinated. Hence third step introduces the coordination service. Step 4 extends a certain model. For instance, there may be a manufacturing service pattern that generates waste and a waste management pattern, then the first one can be extended by merging it with the second. Step 5 is adding enhancing services, such as management services, to other services where necessary. Note that the enhancing services can be decomposed, enhanced, etc. as well. Step 6 does specialization, for instance, from bike to race bike. Step 7 specifically supports model evolution. Model evolution includes outsourcing services, but also reengineering in the form of the reversal of any step 2-6. Our aim is to support meaningful model evolution steps, not arbitrary deletions and insertions.

4 Service Integration

The role of BSP is not limited to design business services. It can also be used in the discovery of the services in a service marketplace or service library. In this section we provide a meta-model for service integration and an example to illustrate the concept.

4.1 Meta-model for Service Integration

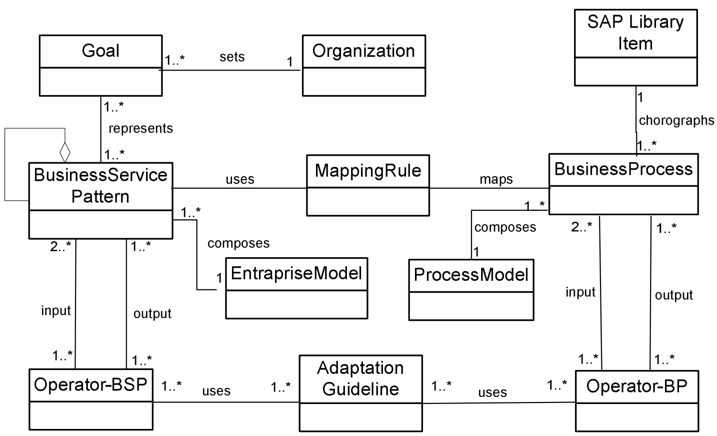


Fig. 6. Framework for pattern-based service integration

Fig. 6 shows the meta-model for composing the enterprise model and how these patterns are related in the service discovery. As soft goals (extra-functional requirements) should be optimized, service discovery should address these goals. Hence these goals are represented in the BSPs: some BSP may prioritize efficiency, another customer-intimacy. Between the BPSs and the business processes a mapping exists, as discussed in our previous work [4]. When using the BSPs in a service marketplace setting, these mappings may also be defined manually. As explained above, BSPs can be combined using algebraic operators, including first of all “merge”. Each time the enterprise model is transformed, using a BSP and BSP operator, the corresponding BPs are integrated as well, so that when finishing the enterprise model, the designer also has an integrated BP (to be more precise: a set of possible BPs as we assume that the enterprise model still leaves room for different process implementations). The adaptation guide lines describe the homeomorphism between BSPs operators and BP operators, including conditions and their use and pragmatic guidelines to the designer.

4.2 Example

The soft goal of customer intimacy can be optimized by delivering the product to the customer at right time. This can be achieved by having an option of outsourcing the delivery service when self delivery is not possible due to lack of resources. This requirement can be implemented by merging the basic “Product Delivery” pattern [Fig. 4(a)] with “Delivery Outsourcing” pattern [Fig. 4(b)]. These two patterns are combined using a merge operator with the common node e.g. delivery. The composite pattern is shown in Fig. 4(c).

The corresponding change at the business process side can be viewed as creating a relationship by means of a message flows (“delivery request”) from the company to the external agent. It is the operator in business process side. Adaptation guidelines:

Rule 1:
For every merge operator in the BSP, there are one or more message flows in the BP.

Conditions:
If the merge operator is based on a service:
- If the service exists internally, the message flows run in between swim lanes of same pool.
- If the service belongs to another party, the message flows run in between swim lanes of different pools.

In the case that the services themselves can be found in the library, or an external market registry, the guidelines provide the requirements to glue them together in coherent processes.

5 Conclusion

In this research work we have presented an efficient modeling method for business services using service patterns. As these patterns are based on the well-established business ontology REA, it provides a truly service-oriented modeling step at the conceptual level. The design is further facilitated by providing a systematic way of using these patterns for pattern composition by means of operators and model completion using design steps. The role of business service patterns is not limited to the design of business services, but does also support service integration at implementation level. The idea of transformation rules aligning the conceptual and PIM/PSM level is core to the MDA approach and not new, but the pattern-based approach extends this idea and makes it more realistic. In the future, we plan to evaluate the proposed method with a more general case and by means of further formal analysis.

References

1. Allgaier, M., Heller, M., Weidner, M.: Towards a Model-based Service Integration Framework for Extensible Enterprise Systems. In: Enterprise Resource Planning und Transformation von ERP-Systemen 1523, MKWI (2010)
2. Cummins, F.A.: BPM meets SOA. In: vom Brocke, J., Rosemann, M. (eds.) Handbook on Business Process Management, vol. 1. Springer, Berlin (2010)
3. Roy, M., Suleiman, B., Weber, I.: Facilitating Enterprise Service Discovery for Non-technical Business Users. In: Maximilien, E.M., Rossi, G., Yuan, S.-T., Ludwig, H., Fantinato, M. (eds.) ICSOC 2010. LNCS, vol. 6568, pp. 100–110. Springer, Heidelberg (2011)
4. Jayasinghe Arachchige, J., Weigand, H., Jeusfeld, M.: Business Service Modeling for the Service-Oriented Enterprise. International Journal of Information System Modeling and Design 3(1) (2012)
5. Zdravkovic, J., Ilayperuma, T.: A Model-driven Approach for Designing E-Services Using Business Ontological Frameworks. In: 2010 14th IEEE International Enterprise Distributed Object Computing Conference (EDOC), pp. 121–130 (2010)
6. Weigand, H., Johannesson, P., Andersson, B., Bergholtz, M.: Value-Based Service Modeling and Design: Toward a Unified View of Services. In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) CAiSE 2009. LNCS, vol. 5565, pp. 410–424. Springer, Heidelberg (2009)
7. De Castro, V., Marcos, E., Wieringa, R.: Towards a service-oriented MDA-based approach to the alignment of business processes with IT-systems: from the business model to a web service composition model. International Journal of Cooperative Information Systems 18(2), 225–260 (2009)
8. Jayasinghe Arachchige, J., Weigand, H.: A Pattern based Approach for Business Service Integration. In: Proc. 6th Int. Workshop on Value Modeling and Business Ontology, VMBO 2012 (2012)
9. McCarthy, W.E.: The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment. The Accounting Review, 554–575 (1982)
10. Bottoni, P., Guerra, E., de Lara, J.: A language-independent and formal approach to pattern-based modelling with support for composition and analysis. Int. Journal Information and Software Technology 52(8) (2010)

Towards Consumer Preference-Aware Requirements

Eric-Oluf Svee, Constantinos Giannoulis, and Jelena Zdravkovic

Stockholm University, Department of Computer and Systems Sciences,
Forum 100, 16440 Kista, Sweden
{eric-sve,constantinos,jelenaz}@dsv.su.se

Abstract. From the business perspective, one of the core concerns within Business-IT alignment is coordinating strategic initiatives and plans with Information Systems (IS). However, while substantial work has been done on linking strategy to requirements for IS development, it has usually been focused on the core value exchanges offered by the business, overlooking other aspects influencing the implementation of strategy. One of these, *consumer preferences*, has been proven to influence the successful provisioning of the business's customer value proposition, and this study aims to establish a conceptual link between them and system requirements. The core contention is that reflecting consumer preferences through business strategy in system requirements allows for the development of systems aligned to consumer preferences, and therefore systems that better support a consumer orientation, where the reasoning behind a particular solution stems from them. The contribution of this paper is the proposal of a consumer preference meta-model along with an illustration of its relationship to a requirements' technique (i*) through the Strategy Maps business strategy formulation.

Keywords: consumer preference, strategy maps, balanced scorecards, i*.

1 Introduction

The term *value proposition* describes how a business will create differentiated, sustainable value for its customers [1]. They are generally expressed quantitatively, as an amount in goods, money, services or rights, considered as a suitable equivalent for something else: a fair price or return for an investment [2]. In counterpoise are qualitative measures, which detail how a good or a service is delivered to, or perceived by, the consumer. These include non-economic values [3], internal values [4], and consumer values [5], among others.

Kotler [6] considered consumer preferences as playing a key role in business; as the key motivator behind, and the primary driver within economic value exchanges, they induce the consumer to seek solutions to fulfill their needs. For the business to deliver on its value proposition by successfully providing those need-fulfilling goods or services which consumers desire in the method and manner which they prefer, it is necessary for it to create a supporting infrastructure, a key component of which often are information systems. Therefore, a model needs be developed capable of capturing consumer preferences, and then presenting these to the business at an appropriate

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 531–542, 2012.

level and in an understandable way, such as through requirements for developing such information systems.

Book selling, a highly dynamic and price-sensitive business, can illustrate the significant impact consumer preferences can have on the IT systems created to deliver the value proposition to customers. *Borders* was the largest book store in the United States, but they missed the shift of the book selling business from physical locations to e-sales, going so far as to completely outsource their online store to *Amazon*. Shopping online instead of in bookstores became appealing to consumers for reasons such as convenience, while the core value exchange remained money for books. However, a brick-and-mortar competitor *Barnes & Noble* did not overlook this shift in consumer preferences and extended its traditional business into the online world while preserving its book stores. The final outcome was that Borders recently went out of business [7].

Similarly, *Amazon* set up its entire business around this shift in consumer preferences, and soon dominated online book selling. For example, to support different consumer preferences surrounding convenience, the company developed entirely new capabilities to purchase and deliver e-books, something for which its infrastructure for processing and shipping physical goods would not have been designed. Again in this example, although the core value exchange remains money for books, the consumer preferences making this exchange appealing vary greatly, and this variability demonstrates how they directly influence the business's success or failure. Success depends on the business's capability to align its IT systems to efficiently marshal and align its resources to aid in effectively presenting, and delivering upon, its value proposition to consumers [8].

The goal of the paper is to relate consumer preferences to system requirements by using the model-driven approach for mapping consumer-related notions to business strategy, and further to the requirements. The proposal is based on three previous studies; first, where it was argued and illustrated how consumer preferences influence business strategy, with the latter being a natural container for capturing those preferences [9]; the second, where a model-level mapping from business strategy to system requirements using the i* technique was presented [10]; and the third, where the enterprise architecture standard ISO 42010 was made consumer-values aware [11].

Section 2 provides a background to research on consumer preferences and briefly introduces business strategy through Strategy Maps and Balanced Scorecards (SMBSC) [1]. Section 3 defines a meta-model for consumer preferences, which in Section 4 is mapped to the meta-model for SMBSC. In Section 5, the impact of the consumer-aware SMBSC on system requirements is demonstrated via i*, based on a business transaction from the book selling business, as well as on the mapping framework proposed in [10]. Section 6 presents conclusions and direction for further work.

2 Background

This section introduces the concept *value*, frameworks and methods for working with it, as well as SMBSC, an established conceptualized business strategy formulation.

2.1 Frameworks for Understanding Consumer Preferences

To clarify the concept of value, conceptual frameworks for its description and discussion, as well as means to measure it, are utilized within this research. There are a number of possibilities to choose from, coming from various fields such as psychology and organizational theory with McLelland's Three Needs Theory [12], and retailing with Parasuraman's ServQual [13], among others. For illustrative purposes, this report relies on three: Maslow's Hierarchy of Needs [14], Schwartz's Value Theory [15], and Holbrook's Typology of Consumer Values [5]. These were selected due to their wide acceptance, application across a variety of industries, and robust conceptual frameworks.

A term that captured both the intentional and evaluative aspects of what drives the value exchange process was sought, with *consumer preference* selected. This was chosen over the more commonly used *customer* to highlight and then reinforce the conceptual break that this research attempts to make; by adopting a term that is not explicitly bound by economic transaction, the focus shifts to the comparative act of consumption, rather than remaining on the level of simple resource exchange.

The choice of consumer preference is grounded in the work of Powell-Mantel et. al., who break down consumer preference into two types: attribute-based, involving comparing brands based on specific attributes, and attitude-based, involving overall evaluations [16]. The term will be used in this work as a rubric for the three primary drivers that cause consumers to seek out goods and services: *Consumer Need*, a basic human need that must be satisfied (derived from psychology via Maslow); *Consumer Motivation*, a belief for what is important in life (coming from psychology via Schwartz); and *Consumer Value*, a judgment based on a comparative, preferential experience (coming from marketing via Holbrook).

Consumer Needs.
Human motivation was explored by Maslow [14] in which he first proposed his *Hierarchy of Needs*. In its final form offered nearly thirty years later, there are seven categories. Beginning with those of a basic necessity then moving to those that are needed for a more fully realized life, these are: *Physiological* (breathing, eating, excreting), *Safety* (security of body, employment, resources, health, property), *Love* (friendship, family), *Esteem* (self-esteem, confidence, achievement), *Cognitive* (knowledge, meaning), *Aesthetic* (appreciation and search for beauty, balance, and form) and *Self-actualization* (realizing personal potential, self-fulfillment).

The *Maslowian Assessment Survey* (MAS), a 195-item, Likert-type instrument offered by Williams and Page [17], is designed to measure three levels of Maslow's Hierarchy in adult populations: safety and security, belongingness and love, and esteem. This relates to Powell-Mantel's attitude-based consumer preference.

Consumer Motivation.
Another type of consumer preference is found in the motivational constructs of Schwartz's *Value Theory* (SVT) [15]. It adopts the definition of value from Rokeach, summarized as a belief that a specific mode of conduct or end-state is personally or socially preferable to its opposite [18]. This relates to Powell-Mantel's attitude-based consumer preference. According to this, values serve as criteria for judgments, preferences, choices, and decisions as they support the person's knowledge, beliefs,

and attitudes. SVT emphasizes the profound nature of values, but at the same time can offer a new consumer research approach by concretely combining these value structures with an analysis of human motivation.

Schwartz claims that items found in earlier value theories, in value questionnaires from different cultures, as well as religious and philosophical discussions of values, can be classified into one of ten motivationally distinct basic values [15]: *Power*, *Universalism*, *Achievement*, *Benevolence*, *Hedonism*, *Tradition*, *Stimulation*, *Conformity*, *Self-direction*, and *Security.*

Schwartz's *Value Survey* (SVS) was developed to measure the basic values laid out in SVT. SVS focuses on a universally applicable method for capturing and describing values across cultures and has been applied in numerous places [15], among which business as well for business strategy development support [19]. The Value Survey operationalizes all ten values with a set of 56 items, and the answers from the questionnaire can be converted into a set of numerical results that can be used directly, or visualized via a value structure.

Consumer Value.

Holbrook's *Typology of Consumer Values* [5] refines the value concept, focusing on those held by individuals during a value exchange, referring to them as consumer values and classifying them into a Typology of Consumer Values.

A consumer value is "an interactive, relativistic preference experience" [5]; interactive entails an interaction between some subject and an object, relativistic refers to consumer values being comparative, preferential refers to consumer values embodying the outcome of an evaluative judgment, and experience refers to consumer values not residing in the product/service acquired but in the consumption experience. Holbrook's definition allows for a rather expansive view of value, because all products provide services in their capacity to create need- or want- satisfying experiences. This related to Powell-Mantel's attribute-based consumer preference.

Three consumer value dimensions are the basis for his typology [5]: *Extrinsic/Intrinsic*, *Self-oriented/Other-oriented*, and *Active/Reactive*. Based on these eight archetypes that represent distinct types of value in the consumption experience—*Efficiency*, *Excellence*, *Status*, *Esteem*, *Play*, *Aesthetics*, *Ethics*, and *Spirituality*—are identified.

In the business-IT alignment discipline, value is most commonly used in an economic sense, to mean an object that can be offered by one actor to another [20] often where the worth or desirability of something is expressed as an amount of money [21]. A value object (also called a resource) is considered as something of economic value for at least one actor, e.g., a car, a book, Internet access or a stream of music [3]. Henkel et.al. state that values can be of more psychological and social natures, such as beauty, pleasure, health state, honor or a feeling of safety [22]. According to Gordijn [23], a user experience is also recognized as having a value. To distinguish between these different kinds of values, Ilayperuma and Zdravkovic identified two categories of values—economic and internal [4] — where internal value could be a certain property attached to an actor, such as their beauty or health, or it could be a property of some enabling service, such as speedy delivery. In contrast to the present proposal, none of these terms from Business-IT alignment functions sufficiently to capture and classify explicitly what constitute the focus within the present work.

2.2 Strategy Maps and Balanced Scorecards (SMBSC)

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. Kaplan and Norton proposed a template for strategy maps representing how an organization can create value. Starting from a mission statement and core values, a strategic vision is defined, which projects the organization's overall goal. 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) [1]. 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.

The figure below presents a conceptualization of SMBSC [10] that supports mappings of consumer preferences to four generic types of customer value proposition: *Best Total Cost*, *Product Leader*, *Complete Customer*, and *System Lock-In* [1]. These four generic types of customer value proposition proposed by Kaplan & Norton frame the set of generic goals set in the Customer Perspective

3 Consumer Preference Meta-model (CPMM)

A meta-model for expressing consumer preferences—the Consumer-Preference-aware Meta-model (CPMM)—is now proposed. After its explication, SMBSC—a meta-model capable of a business strategy is then related to it.

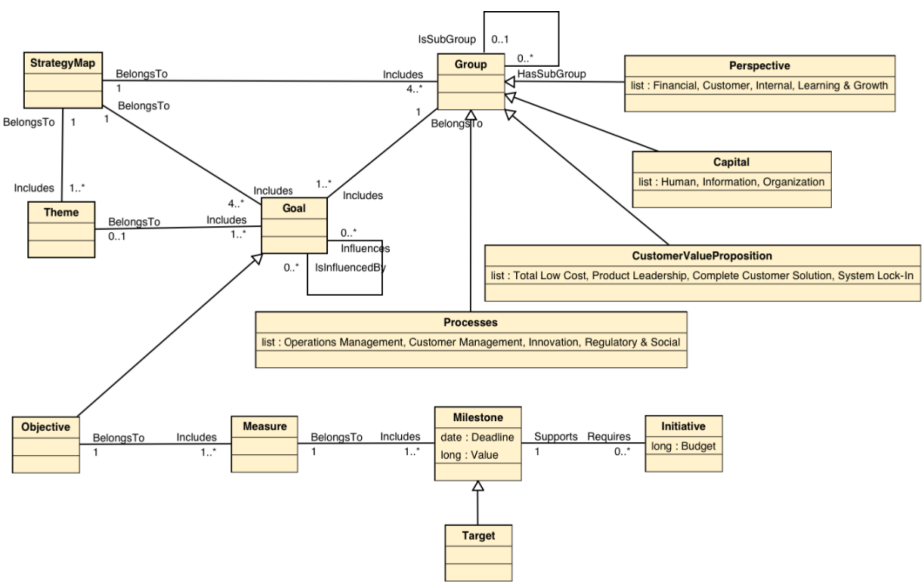


Fig. 1. SMBSC Meta-model [10]

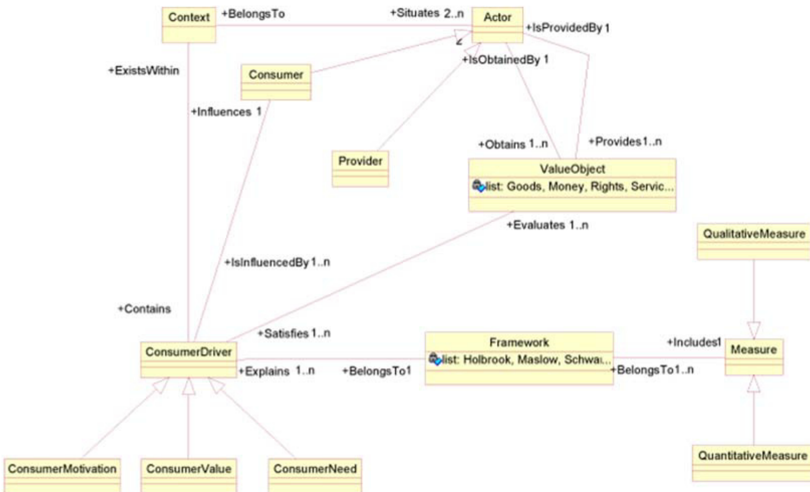


Fig. 2. Consumer Preference Meta-Model (CPMM)

Class Descriptions.
CPMM.Context represents the premises within which a transaction takes place (i.e. it may refer to location, either physical or virtual). For the bookselling scenario, an example of context would be an online store.

CPMM.Actor contains both *CPMM.Consumer* and *CPMM.Provider*, economically independent entities that are the primary participants within the exchange of goods, money, rights, or services. In the example, *CPMM.Consumer* is the consumer, the *CPMM.Provider* is the bookseller.

CPMM.ValueObject is the focus of the process wherein the consumer evaluates whether the value object satisfies the motivation, value or need driving their desire to participate in the exchange process. In the bookseller example, this would be a book.

CPMM.Framework provides a theoretical means to understand and explain consumer drivers that need to be satisfied through the consumption experience. In the bookseller example, a marketing survey reports that customers want to be actively engaged in an enjoyable shopping experience. An example of this is Maslow's Hierarchy.

CPMM.Measure quantifies and conceptualizes values. Its sub-classes– *CPMM.QualitativeMeasure* and *CPMM.QuantitativeMeasure*– are means contained within each of the various frameworks that can be used for conceptualization and quantification. An example of *CPMM.QualitativeMeasure* would be the core dimensions of Holbrook's Typology when used for classifying and explaining a consumer preference. *CPMM.QuantitativeMeasure* is exemplified via Schwartz's Value Survey [15] where the results of the questionaire can provide a value profile.

CPMM.ConsumerDriver is the preferences that drive a consumer's evaluative process as they seek fulfillment. Self-actualization is a consumer driver that could lead someone to purchase a self-help book from a book seller to aid their personal growth.

CPMM.Maslow:Framework contains Maslow's Hierarchy (Physiological, Safety, Love, Esteem, Cognitive, Aesthetic, Self-actualization).

CPMM.Holbrook:Framework, contains Holbrook's Typology (Efficiency, Excellence, Status, Esteem, Play, Aesthetics, Ethics, Spirituality).

CPMM.Schwartz:Framework, contains Schwartz's Values (Power, Universalism, Achievement, Benevolence, Hedonism, Tradition, Stimulation, Conformity, Self-direction, Security, and Spirituality).

Constraints.
Apart from the cardinality constraints included in the meta-model, a set of constraints is also introduced to capture the permissible instantiations of concepts found in the frameworks.

At least one instance of both *CPMM.Consumer* and *CPMM.Provider* must belong to the same *CPMM.Context.*

Moreover, an instance of *CPMM.ValueObject* provided by an instance of *CPMM.Provider* which is an instance of *CPMM.Actor* which belongs to an instance of *CPMM.Context* is the same instance of *CPMM.ValueObject* obtained by an instance of *CPMM.Consumer* which is an instance of *CPMM.Actor* that belongs to the same instance of *CPMM.Context.*

The frameworks of Maslow and Schwartz possess quantitative measures, whereas only Holbrook's includes qualitative measures. *CPMM.QuantitativeMeasure* is a *CPMM.Measure* that *BelongsTo* *CPMM.Framework:Schwartz* or *CPMM.Framework:Maslow*. *CPMM.QualitativeMeasure* is a *CPMM.Measure* that *BelongsTo* *CPMM.Framework:Holbrook*.

Furthermore, each sub-class of *CPMM.ConsumerDriver* is related to a particular framework. *CPMM.ConsumerDriver.ConsumerNeed* can only *BelongTo* an instance of *CPMM.Framework.Maslow*, *CPMM.ConsumerDriver.ConsumerMotivation* class can only *BelongTo* an instance of *CPMM.Framework.Schwartz*, and similarly an instance of *CPMM.ConsumerDriver.ConsumerValue* can only *BelongTo* an instance of *CPMM.Framework.Holbrook*.

3.1 Relating CPMM to SMBSC Meta-model

SMBSC includes four generic types of customer value proposition: *Best Total Cost*, *Product Leader*, *Complete Customer*, and *System Lock-In* [1], which consequently frames a set of generic goals under each customer value proposition type. Based on this grouping of goals and their mapping to consumer drivers as captured by CPMM, the Customer Perspective of SMBSC reflects customer preferences in a traceable manner. In Table 1, consumer drivers captured by *CPMM.ConsumerDriver* are related to the generic strategic goals of the four types of customer value proposition of SMBSC captured by *SMBSC.CustomerValueProposition*.

Generic strategic goals for Best Total Cost aim at consistent, timely and low-cost offerings [1], where goals on (i) lowest-cost suppliers, aiming at attractive prices, (ii) consistently high quality, aiming at excellent and consistent quality offering, (iii) speedy purchase, aiming at short lead times and ease of purchase, and (iv) appropriate selection, aiming at good selection offering. Following the mappings of Table 1 where generic strategic goals are mapped to consumer preferences, strategic goals can be linked to particular consumer drivers (need, motivation, value).

Considering the bookselling scenario, where a bookseller adopts the *Best Total Cost* customer value proposition, the strategic goal "*Select Lowest-Cost Supplier"* is an instance of *SMBSC.Goal*, which *BelongsTo SMBSC.CustomerValue Proposition:BestTotalCost*, which *IsSubGroupOf SMBSC.Perspective:Customer* which is a *SMBSC.Group*. Such a goal can be measured, thus is an instance of *SMBSC.Objective* and *Includes* a *SMBSC.Measure* captured by the cost offerings provided from suppliers, expressed in a monetary unit. According to Table 1, the *SMBSC.Goal* "*Select Lowest-Cost Supplier"* of the bookseller can be linked to following consumer drivers: *Safety*, *Security*, *Efficiency*, and *Excellence*.

Safety expresses security in resources and property. The bookseller must ensure that selecting the price being offered is the lowest among suppliers to constitute a safe investment. Ensuring the lowest price offered among suppliers is an instance of *CPMM.ConsumerNeed:Safety* which is a *CPMM.ConsumerDriver* and *BelongsTo CPMM.Framework:Maslow* that provides both the quantitative measure of a value, but also the conceptual framework to explain it.

Security expresses stability in relationships. The bookseller must ensure that selecting the supplier offering the lowest price constitutes a secure investment. Ensuring a secure investment among low-cost suppliers is an instance of *CPMM.ConsumerMotivation:Security* which is *a CPMM.ConsumerDriver* and *BelongsTo CPMM.Framework:Schwartz* that provides both the quantitative measure of a value, but also the conceptual framework to explain it.

Efficiency is often expressed as a ratio of inputs versus outputs. For the bookseller efficiency could be referring to a supplier that allows minimal shipments to customers. Choosing a low-cost supplier offering efficient shipments is an instance of *CPMM.ConsumerValue:Efficiency* which is a *CPMM.ConsumerDriver* and *BelongsTo CPMM.Framework:Holbrook* that provides both the qualitative means to both express and to measure it.

Excellence expresses an appreciation of something's potential ability to serve as a means to some end. The bookseller must ensure that when selecting the supplier offering the lowest price, reputation is considered. Ensuring supplier's reputation is an instance of *CPMM.ConsumerValue:Excellence* which is a *CPMM.ConsumerDriver* and *BelongsTo CPMM.Framework:Holbrook* that provides both the qualitative means to both express and to measure it.

Each instance of the aforementioned consumer drivers evaluates the same instance of *CPMM.ValueObject:Goods* which is book supplies, where each supplier is an instance of *CPMM.Provider* which is a *CPMM.Actor* and the bookseller is an instance of *CPMM.Consumer* that *IsInfluencedBy* all aforementioned *CPMM.ConsumerDriver*.

Additionally, *SMBSC.Measure* which captures the means to evaluate the achievement of a *SMBSC.Objective* which is a measureable goal, thus captures a quantifiable measure can be associated with *CPMM.Framework.QuantiativeMeasure* which also captures quantifiable measures.

Similar instantiations can be derived for each generic strategic goal of the four types of customer value propositions of SMBSC based on the mappings of Table 1.

Table 1. Strategic Goals Related to Consumer Drivers

Goals from Strategy Map Templates	Consumer Drivers		
	Consumer Need	**Consumer Motivation**	**Consumer Value**
Best Total Cost			
Lowest Cost Supplier	Safety	Security	Efficiency Excellence
Consistently High Quality	Aesthetic	Conformity, Security, Power, Hedonism, Tradition	Efficiency, Excellence
Speedy Purchase	Safety	Self-direction	Efficiency, Excellence
Appropriate Selection	Safety	Self-direction, Spirituality	Efficiency, Excellence
Product Leader			
High Performance Products	Esteem, Aesthetic	Conformity, Security, Power, Stimulation, Hedonism	Efficiency, Excellence, Play, Aesthetics
First to Market	Safety, Love, Esteem, Aesthetic, Self-actualization	Self-direction, Stimulation	Efficiency, Excellence
New Customer Segments	Safety, Love, Esteem, Aesthetic, Self-actualization	Stimulation, Tradition, Universalism, Spirituality	Efficiency, Excellence, Play, Aesthetics, Esteem Status, Ethics, Spirituality
Complete Customer Solutions			
Quality of Solutions Provided	Safety	Conformity, Security, Hedonism, Tradition, Power, Spirituality	Efficiency, Excellence, Play Aesthetics Esteem
Number of Products/ Services per Customer	Safety, Esteem	Self-direction	Efficiency, Excellence
Customer Retention	Safety, Love, Esteem, Aesthetic, Self-actualization	Conformity, Security, Tradition, Universalism, Spirituality	Efficiency, Excellence, Play, Aesthetics, Esteem, Status, Ethics, Spirituality
Lifetime Customer Profitability	Safety, Love, Esteem, Aesthetic, Self-actualization	Tradition, Spirituality	Efficiency, Excellence, Play, Aesthetics, Esteem
System Lock-in			
Broad Selection/ Convenient Access	Safety, Esteem	Self-direction, Security	Efficiency, Excellence
Widely Used Standard	Safety, Esteem	Conformity, Security, Achievement, Tradition, Universalism	Efficiency, Excellence
Stable Platform Innovation	Safety	Achievement, Tradition, Universalism	Efficiency, Excellence
Large Customer Base	Safety	Achievement, Tradition, Universalism	Efficiency, Excellence
Easy-to-Use Platform and Standard	Safety	Self-direction, Tradition, Universalism	Efficiency, Excellence

The linkages identified between generic strategic goals (SMBSC) and consumer preferences (CPMM) allow enriching the former with a set of goals focused on capturing consumer preferences, thus increasing the level of detail of SMBSC. This means that the generic strategic goal "*Select Lowest-Cost Supplier*" is influenced by goals on safety, security, efficiency and excellence aiming at satisfying the need, the motivation and value of a consumer.

For the bookselling scenario, based on the linkages identified earlier, the *SMBSC.Goal "Select Lowest-Cost Supplier" BelongsTo SMBSC.CustomerValueProposition:BestTotalCost* which *IsSubGroupOf SMBSC.Perspective:Customer:*

- *IsInfluencedBy SMBSC.Goal "Ensure lowest price offered"* which *BelongsTo SMBSC.CustomerValueProposition:BestTotalCost* which *IsSubGroupOf SMBSC.Perspective:Customer,*
- *IsInfluencedBy SMBSC.Goal "Ensure secure investment among suppliers"* which *BelongsTo SMBSC.CustomerValueProposition:BestTotalCost* which *IsSubGroupOf SMBSC.Perspective:Customer,*
- *IsInfluencedBy SMBSC.Goal "Choose supplier offering efficient shipments"* which *BelongsTo SMBSC.CustomerValueProposition:BestTotalCost* which *IsSubGroupOf SMBSC.Perspective:Customer,*
- *IsInfluencedBy SMBSC.Goal "Ensure supplier's reputation"* which *BelongsTo SMBSC.CustomerValueProposition:BestTotalCost* which *IsSubGroupOf SMBSC.Perspective:Customer.*

4 Consumer Preference: From Strategy to Requirements Using i*

Relating consumer preference with SMBSC by introducing strategic goals that express consumer drivers allows the use of the model-level mapping of SMBSC concepts to system requirements using i* [10]. i* is a goal modeling technique used in requirements engineering operationalizing goals through concrete actions and design decisions [24]. Based on the mappings in [10], the SMBSC strategic goals introduced via consumer drivers of CPMM can be represented as goals or soft-goals in i* allowing the derivation of system requirements that support their operationalization.

Since consumer preference is strongly related to customer value proposition, mappings to i* are relevant only to the customer perspective of SMBSC. Additionally, strategic goals originating from consumer needs or consumer motivations are represented as goals in i* because the frameworks they belong to use quantitative measures (see CPMM). Strategic goals originating from consumer values are represented as soft-goals in i* because the framework they belong to uses qualitative measures (see CPMM).

For the bookselling scenario, Figure 3, presents a partial i* SRM of the SMBSC strategic goal "*Select Lowest-Cost Supplier*" including consumer preference. The goals derived are based on linkages to consumer drivers, e.g. "Lowest Priced Offer to be Ensured" and are accompanied with tasks pointing towards actions for their achievement through *Means-ends* links. Thus, pointing to their operationalization through concrete actions and design decisions for a system that should compare all prices offered, evaluate suppliers' trustworthiness, shipment policies and reputation.

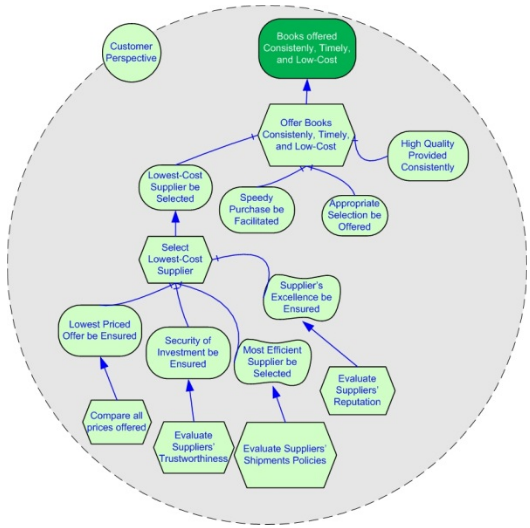


Fig. 3. A bookseller's Strategic Goals on Consumer Preference illustrated in i*

5 Conclusions and Future Work

In this study a Consumer Preference Meta-Model (CPMM) has been presented to establish linkages between consumer preferences and generic strategic goals formulated by SMBSC. Their purpose is to reflect consumer preferences through business strategy to system requirements allowing for the development of systems that better support a business's customer value proposition. Establishing such linkages provides traceability between consumer preferences and system requirements through business strategy. This allows consumer preferences to appear as stakeholder requirements (via SMBSC mappings to i* in a traceable manner through strategy.

The goal of the paper was to establish linkages between consumer preferences and business strategy to illustrate how consumer preferences can be reflected in requirements. The former has been achieved by modeling consumer preferences in CPMM and arguing for the relationships of consumer drivers to strategic goals of SMBSC based on four generic customer value propositions. The latter has been illustrated by using a model-level approach for mapping SMBSC concepts to i* constructs constituting transparent/clear what consumer preference strategic goals in the consumer perspective in SMBSC are directed towards. Throughout the paper, a bookselling scenario has been used to explain and illustrate the applicability of the proposal.

Future directions of the work are focused on applying a complete case to evaluate the presented meta-model, as well as the traceability of consumer preferences to requirements derived through SMBSC.

References

1. Kaplan, R., Norton, D.: Strategy Maps: Converting Intangible Assets into Tangible Outcomes. Harvard Business Review Press (2004)
2. McCarthy, W.E.: The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment. The Accounting Review (1982)

3. Afuah, A., Tucci, C.: Internet Business Models & Strategies. McGraw Hill, Boston (2003)
4. Ilayperuma, T., Zdravkovic, J.: Exploring Business Value Models from the Inter-Organizational Collaboration Perspective. In: ACM Symposium on Applied Computing (SAC) Sierre, Switzerland, pp. 99–105 (2010)
5. Holbrook, M.B.: Consumer Value: A Framework for Analysis and Research. Routledge, London (1998)
6. Kotler, P.J.: Marketing Management, 7th edn. Prentice-Hall, Englewood Cliffs (1991)
7. The New York Times (January 25, 2012), `http://nyti.ms/oThCIm`
8. Inside Retailing (April 9, 2012), `http://www.insideretailing.com.au/IR/IRnews/How-Amazon-is-redefining-the-book-market-4630.aspx`
9. Svee, E.O., Giannoulis, C., Zdravkovic, J.: Modeling Business Strategy: A Consumer Value Perspective. In: Johannesson, P., Krogstie, J., Opdahl, A.L. (eds.) PoEM 2011. LNBIP, vol. 92, pp. 67–81. Springer, Heidelberg (2011)
10. Giannoulis, C., Zdravkovic, J.: Modeling Strategy Maps and Balanced Scorecards using i*. In: 5th Int'l i* Workshop (iStar 2011), vol. 766, pp. 90–95. CEUR-WS, Aachen (2011)
11. Svee, E.O., Zdravkovic, J., Giannoulis, C.: Consumer value-aware enterprise architecture. In: Venkatraman, B.I.N. (ed.) ICSOB 2012. LNBIP, vol. 114, pp. 55–69. Springer, Heidelberg (2012)
12. McClelland, D.C.: Human Motivation. Cambridge University Press (1988)
13. Parasuraman, A., Zeithaml, V., Berry, L.: SERVQUAL: A Multiple-Item Scale for Measuring Consumer Perception of Service Quality. J. Retailing 64, 12–40 (1988)
14. Maslow, A.H.: A Theory of Human Motivation. Psychological Rev. 50(4), 370–396 (1943)
15. Schwartz, S.H.: Universals in the Content and Structure of Values: Theoretical Advances and Empirical Tests in 20 Countries. In: Zanna, M. (ed.) Advances in Experimental Social Psychology, San Diego, vol. 25 (1992)
16. Powell-Mantel, S., Kardes, F.J.: The Role of Direction of Comparison, Attribute-Based Processing, and Attitude-Based Processing in Consumer Preference. J. Consumer Research 25, 335–352 (1999)
17. Williams, D., Page, M.M.: A multi-dimensional measure of Maslow's hierarchy of needs. Journal of Research in Personality 23, 192–213 (1989)
18. Rokeach, M.: The Nature of Human Values. The Free Press, New York (1973)
19. Epstein, M., Manzoni, J.F.: Implementing corporate strategy—from Tableaux de Bord to Balanced Scorecards. European Management J. 16(2), 190–203 (1998)
20. Weigand, H., Johannesson, P., Andersson, B., Bergholtz, M., Edirisuriya, A., Ilayperuma, T.: On the Notion of Value Object. In: Martinez, F.H., Pohl, K. (eds.) CAiSE 2006. LNCS, vol. 4001, pp. 321–335. Springer, Heidelberg (2006)
21. Wieringa, R., Gordijn, J., van Eck, P.: Value-Based Business-IT Alignment in Networked Constellations of Enterprises. In: 1st Int'l Workshop on Requirements Engineering for Business Need and IT Alignment (REBNITA 2005), Paris, France (2005)
22. Henkel, M., Johannesson, P., Perjons, E., Zdravkovic, J.: Value and Goal Driven Design of E-Services. In: IEEE Int'l Conf. on E-Business Engineering (ICEBE 2007). IEEE Computer Society Press, Hong Kong (2007)
23. Gordijn, J., Akkermans, H., van Vliet, H.: Business Modelling is Not Process Modelling. In: Mayr, H.C., Liddle, S.W., Thalheim, B. (eds.) ER Workshops 2000. LNCS, vol. 1921, pp. 40–51. Springer, Heidelberg (2000)
24. Yu, E.: Modeling Strategic Relationships for Process Reengineering. PhD. thesis. Department of Computer Science, University of Toronto, Canada (1995)

Extending the REA-DSL by the Planning Layer of the REA Ontology

Dieter Mayrhofer and Christian Huemer

Business Informatics Group, Vienna University of Technology, Austria
{mayrhofer,huemer}@big.tuwien.ac.at

Abstract. The Resource-Event-Agent (REA) ontology is a powerful and well accepted approach towards the design of accounting information systems (AIS). However, the REA notation - that is currently based on class diagrams - is not very intuitive for business experts. Accordingly, we aim at a REA domain specific modeling language that facilitates the communication between business experts and IT professionals. In previous work we defined the REA-DSL operational layer reflecting actual business events which "have occurred" or "are occurring". In this paper we extend the REA-DSL by the planning layer capturing what future events "are scheduled" or "are planned" by commitments. Now, our REA-DSL covers all basic concepts to describe a full accounting infrastructure. The REA-DSL may serve as a solid basis for generating a conceptual AIS data model - which is subject to future work.

Keywords: REA, business models, domain-specific language.

1 Introduction

According to Romney and Steinbart an accounting information system (AIS) is a system that collects, records, stores, and processes financial and accounting data to produce information for decision makers [1]. They define the following main functions of an AIS: (i) collect and store data about events, resources, and agents; (ii) transform that data into information that management can use to make decisions about events, resources, and agents; and (iii) provide adequate controls to ensure that the entity's resources including data are available when needed as well as accurate and reliable. Accordingly, financial and accounting related data is analyzed, prepared, calculated, and visualized in order to provide insight into a company's current financial status as well as of its historic activities. Furthermore, the data helps to predict the financial future of a company and, thereby, helps managers in their decision making.

In order to create a useful and meaningful AIS, the data structure and user interface has to reflect the economic phenomena on which companies base their business. Thus, it is essential that the business people providing the requirements can unambiguously communicate with the IT professionals which are in charge of creating the AIS. Business ontologies - providing abstract descriptions of enterprises in their business context - may be used as a language for communicating these requirements between domain experts and IT staff.

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 543–554, 2012.

The most prominent business ontology for accounting information systems is the Resource-Event-Agent (REA) ontology developed by McCarthy, Geerts and others [2]. REA is a widely accepted framework for the design of a conceptual model of the accountability infrastructure of enterprise information systems. Originally, REA targeted the resource flows within and between companies describing what is currently occurring and what has occurred in the past. This is known as the operational layer. Later it was extended by a planning layer and a policy layer capturing what should, could, or must be occurring sometime in the future [3,4].

Today, REA may be considered as a powerful business ontology capturing all relevant data to generate the conceptual design of an AIS. However, we feel that it does not deliver an appropriate representation of the business model which can be understood not only by the IT expert, but also by the business expert. Thus, the use of REA in the design of AIS does not yet reach its full potential. We argue, that an easy-to-understand REA notation will accelerate, streamline, and reduce the costs of the AIS development process.

Consequently, we started the endeavor of developing a domain specific modeling language for the REA concepts which aims at both (i) delivering an intuitive REA notation and (ii) retaining the full expressiveness of the REA concepts. Similarly to the development of REA itself, our REA-DSL started on the operational layer only [5]. In this paper, we are going to extend the REA-DSL by concepts of the planning layer and types of the policy layer. Once we have consider all REA layers, we are able to derive a conceptual data model, e.g. an entity-relationship diagram, for the resulting AIS. However, this transformation process is out of scope for this paper and will be addressed in future work.

The remainder of this paper is structured as follows: In Section 2 we give a brief introduction of the REA ontology with focus on the commitment and type artifacts. An example of the REA-DSL is given in Section 3. Section 4 defines the extended REA-DSL. The evaluation is provided by our tool in Section 5 and we conclude the paper by the summary in Section 6.

2 Related Work on REA

When developing a domain specific modeling language for REA, related work focuses more or less only on papers related to REA. A comparison with alternative approach for a domain specific modeling language is impossible, since best to our knowledge no such efforts have been published. Accordingly, we concentrate in the following on REA and in particular on the REA planning layer and types of the policy layer.

REA was originally proposed as a reference framework to conceptualize the resource flows within and between firms in terms of what is currently occurring or what has occurred in the past [2]. Accordingly, REA focuses on the economic exchanges as the central unit of analysis. Instead of representing these exchanges with double-entry bookkeeping artifacts (e.g. debits, credits, accounts), REA proposes concepts and patterns to derive semantic models of economic exchanges and transformations [6].

Following its name, REA is based on the three main concepts: economic resources, economic events, and economic agents. It should be noted that for a better readability we drop the prefix term *economic* for the remainder of this paper. Basically, one or more resources are exchanged between usually two (but in theory also more) agents at well defined events. A cornerstone of REA is also the concept of duality, which means that usually one event (or in theory a set of events) is compensated by another event (or set of events). An example on the instance level may be: On the 30 November 2011 a `sale` (*event*) occurs, where the `salesman Joe` (*agent*) with the help of the `shop assistants Mary` and `Wendy` (*agents*) give `50 pounds of tuna fish` (*resource*) and a `fishing rod` (*resource*) to their `customer Fred` (*agent*). The `sale` (*event*) is compensated by the `payment` (*event*) which happens when `Fred` (*agent*) pays the amount of `700 Euros` to the `cashier Mark` (*agent*).

REA does not model the individual instance on M0, but analyses an enterprise on the model layer M1. Accordingly, the REA model defines the schema for the above mentioned instance from the perspective of the `seller` using the concepts of resource, events, agents, duality, stockflow, and participation. The relationships between these concepts are formalized on the M2 layer as a meta model which we already published at CAiSE 2011 [5].

The above mentioned example demonstrates that the original REA [2,7,6] is capable to record events that have already happened. As mentioned earlier, AIS should provide management an insight into a company's current financial status in order to make proper decisions. Usually, the financial status depends not only on the past, but also on commitments that the company has already made for the (near) future. Thus, it is desirable to extend REA by such commitments and internal plans to fulfill these commitments. Accordingly, REA should be able to capture the following example of a reservation: On 2 November 2011 the `salesman Joe` *commits* himself - as a particular `salesman` - and two (not yet known by name) `shop assistants` to give `customer Fred` at least `40 pounds of tuna`. In return, `customer Fred` *commits* himself to pay a (not yet known by name) `cashier`. It is easy to recognize, that the above mentioned actualization of the events is in-line with the commitments, even if the actualization overfulfills the commitment by additional `10 pounds of tuna` and the `fishing rod`.

In order to accommodate such a scenario REA has been extended by a planning and policy layer [3,8,4]. These layers cover economic activities that should, could, or must happen in a company and are, thus, relevant for planning and controlling activities. The extension allows to model the previous commitment example on an abstract level on M1. Therefore, new concepts are introduced on the meta model layer M2. The most central concept is a legal *commitment*. The concept *commits* associates the *agent* who does the *commitment* to the *commitment*. *Reciprocity* is a relationship between *commitments*. The concept of *fulfill* links *commitments* and associated *events*. *Reserve* is used to associate *resources* or *agents* to *commitments*. Furthermore, REA introduces the *typification* to support the planning and policy layer. From the above example it is evident that in some cases one may already refer to a particular, yet-known *agent* (such as the `salesman Joe`), and in other cases one may only refer to the *type of agent* (such as `shop assistant`),

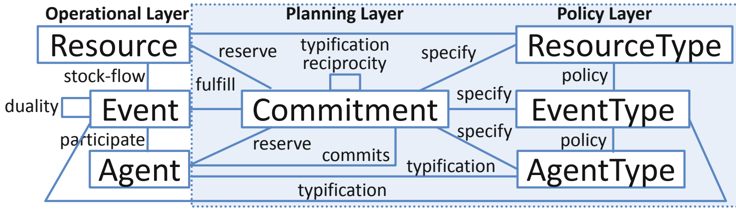


Fig. 1. REA Meta Model

because it is not-yet-known in person. This is reflected by the M2 concept of *agent type*. Corresponding concepts exist for *event types* and *resource types*. An overview of the REA M2 meta level concepts is presented in Figure 1.

Accordingly, we aim at supporting all these concepts in our REA-DSL. Similarly as the papers on the planning and policy layer [3,8,4] extend the basic REA concepts [2,7,6], this paper extends our first paper on the basic REA concepts [5]. In addition, our work in this paper was influenced by Gailly et al. [9]. They already tried to formalize types and commitments but did not provide a domain specific language and separate views for it.

3 REA-DSL Example

Before going into all the theoretical details when explaining the meta model of our REA-DSL, we start illustrating our results by a simple, but still realistic example. The example in this paper is based on *Sy's Fish* - an example that was used also by Geerts at al. [7] to demonstrate REA.

Example Business Model. *Sy's Fish* purchases fishes from the fish market. He sells three different kind of fishes: carp, trout, tuna. Additionally, he acquires products from a factory: books and fishing rods. The fishes and products are transported to *Sy's Fish* by leased trucks. At *Sy's Fish* all the fishes get cleaned. The products and fish are then sold to customers in order to make profit. To be able to accomplish all tasks, *Sy's Fish* employs a couple of employees, which can either be a salesman, shop assistant, or cashier.

Modeling REA. The REA-DSL consists of five different interlinked views: (i) *agents view*, (ii) *resources view*, (iii) *value chain view*, (iv) *planning view*, and (v) *operational view*. You might start modeling by using any of the views. In the following, we provide an example of our REA-DSL.

Agents View. We start modeling the business model with the REA-DSL by defining the different agents. In the agents view (cf. Figure 2a) the agents inside the company are depicted on the left side with the white head stick figures. The general agent is an `employee`, which can either be a `salesman`, a `shop assistant`, or a `cashier`. The outside agents are depicted as black head stick figures. *Sy's Fish* does business with the outside agent `customer`. For the moment we forget about other outside agents which are involved in the overall process, but are not relevant for understanding the example.

Resources View. Next, we define all resources which need to be tracked and recorded in *Sy's Fish* company. These resources are depicted by the shape of a drop in Figure 2b. Resources which can be identified individually are marked by a solid drop, whereas bulk resources where the individual real world objects cannot (or need not to) be identified are marked by dashed drops. Accordingly, the resource `product` is a solid drop. Products can be categorized into `books` and `fishing rods`. Another resource is `cash`. `Cash` is a bulk resource that appears as a dashed drop. Evidently, *Sy's Fish* cannot track each single coin or bill and therefore, only the whole amount of `cash` is of interest. Another bulk resource is `fish`. `Fish` can be categorized into `carp` and `trout`.The last resource is `truck`, which is modeled as a solid drop.

Value Chain View. In the next step a high level overview of *Sy's Fish* business model is provided by depicting it in a value chain view (cf. Figure 3). The value chain contains economic activities that create higher value by value transfers with external partners or transformations inside the company. Usually, resources (which are referenced in the resource view) created by one value activity serve as input to another one. In order to keep Sy's Fish value chain as simple as possible we omit to depict the resource labor which is naturally input to all economic activities.

`Fish` and `products` are purchased with `cash` in the two value activities `fish buying` and `product buying`. The `products` and `fish` are `transported` to *Sy's Fish* by a `truck` leased for `cash` in the `truck acquisition` value activity. The `fish` is cleaned in the `cleaning` value activity and then the `products` and `fish` are sold for `cash` in the `selling` value activity.

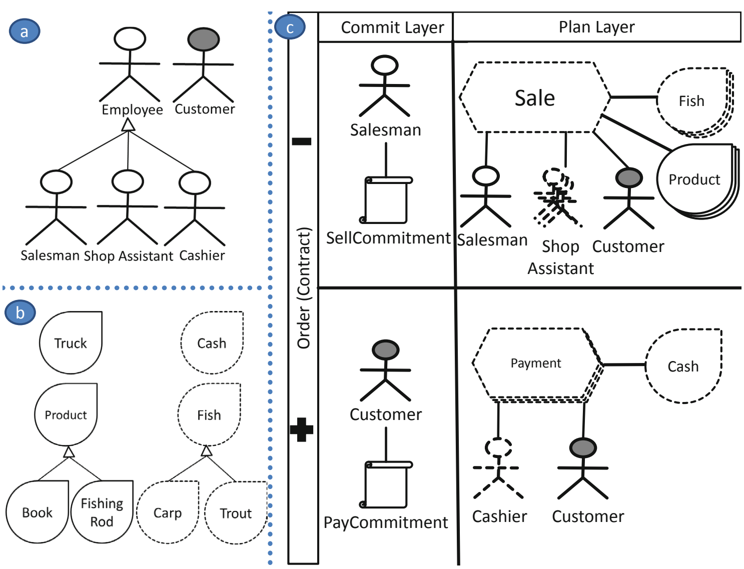


Fig. 2. (a) Agent View, (b) Resource View, and (c) Planning View

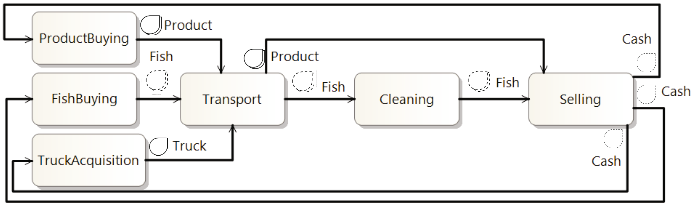


Fig. 3. Value Chain View

Planning View. Each value activity defined in the value chain view can be refined in a referenced planning view. For this example we just elaborate on the `selling` value activity which is depicted in Figure 2c. It also references agents and resources defined in the agents and resources view. The resulting contract is a 2 x 2 matrix. The top covers a commitment leading to a decrease in resources, and the bottom defines the compensating commitment leading to an increase in resources. Orthogonally, the left hand side defines who does the commitment and the right hand side defines what it is about and who is going to fulfill it.

The contract is an `order` and contains two commitments: the `salesman` committing in the `sell` commitment to engage in the `sale` event in the future and the `customer` committing in the `pay` commitment to engage in the `payment` events in the future. A concrete `salesman` will participate in the future `sale` of the resources `fish` and `product`. Since more than one `fish` and more than one `product` are potentially sold, their shapes appear as *stacks of drops*. As mentioned before `fish` are bulk resources with dashed drops and `products` are individually identifiable with *solid drops*. Note, a solid drop in the commitment means also that the individual `product` being part of the future `sale` is exactly defined at the time of the commitment. If only the type of the `product` is specified in the commitment it has to appear a *dashed drop* (or *stack of dashed drops* in case of multiple `products`). Furthermore, the `salesman` will be supported by one or more `shop assistants` (depicted as a stack of dashed stick figures) which are not known at the time of the commitment. These `shop assistants` will get concrete when the `sale` event happens. In the reciprocal commitment, the `customer` commits himself to pay in `cash` to a not-yet-known `cashier`. The staple of `payment` events signifies that there is the option of multiple (partial) `payments`.

In this example we do not further elaborate on the operational view which was the main focus of our CAiSE 2011 paper [5]. The interested reader may refer to this paper. However, it should be noted that one may semi-automatically transfer the right hand side of the contracts in the planning view to the operational view.

4 REA-DSL Formalization

Having introduced Sy's Fish example, we are now elaborating on the formal concepts which this example is based upon. These concepts are represented in the REA-DSL meta model on the M2 layer. We concentrate on the DSL meta model of the planning view. Due to space limitations we do not describe the meta model

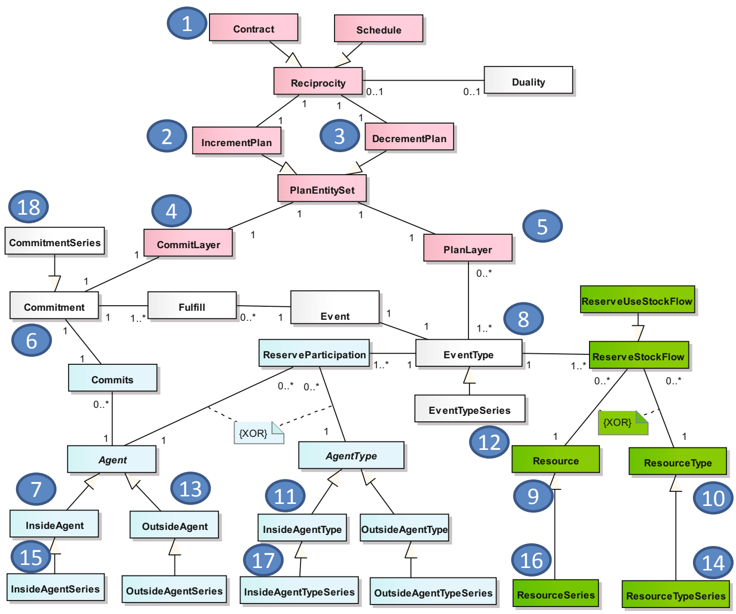


Fig. 4. Commitment Meta Model

of the agent view as well as the the resource view, the operational view, and value chain view, which we slightly extended for the incorporation of types but besides that already covered in [5].

In [5] we have described *exchanges* including *events* in the *operational view*. These exchanges often do not happen unexpected. There can be *commitments* to fulfill events in the future. Such commitments can for example be an `order` to `buy fish` or a schedule to `clean fish`. Thus, events happening in the future are planned beforehand. The major contribution of this paper is to add this commitment concept together with types to the REA-DSL by providing an additional view called the *planning view*. This view makes use of resources and agents defined in the resource and agent view. Following we propose the *planning view* meta model (cf. Figure 4) and its concrete syntax (cf. Figure 5). Classes in the meta model which also have a corresponding stencil in the concrete syntax of the abstract model are marked with a numbered circle.

Reciprocity. The root element of the planning view is the *reciprocity*. A reciprocity is analog to the *duality* in an operational view. The duality connects the incrementing entities with the decrementing entities. Likewise, reciprocity connects the *incrementing commitments* with the *decrementing commitments* in the planning view. The reciprocity may be associated with a duality which refers to the corresponding duality in the operational view and vice versa. Thus, it is the link between the planning view which plans future events and the operational view which shows the events at the time of execution. Notice, if the reciprocity is conceptually congruent to the duality, the duality does not have to be modeled separately.

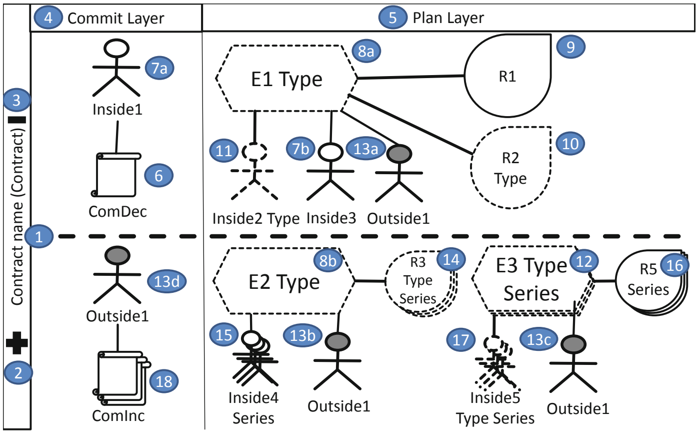


Fig. 5. Commitment Abstract Example

Reciprocity can either be a *contract* (1) or a *schedule*. The abstract example shows a contract (1) (indicated by `contract` in the brackets) with the name `contract name`. In the case of a schedule the term in the brackets would be `schedule`. If the reciprocity is a contract the corresponding duality in the operational view has to be a transfer and if it is a schedule the corresponding duality has to be a transformation, respectively.

Increment Plan and Decrement Plan. The reciprocity consists of two *plan entity sets* called *increment plan* (2) and *decrement plan* (3). The increment plan is depicted as the lower swimlane annotated by a *plus sign* and contains all the commitments and related event types which will lead to an increment of resources in the future. Accordingly, the decrement plan is depicted as a swimlane annotated by a *minus sign* and contains all the commitments and event types which will lead to an decrement of resources in the future.

The *increment plan* (2) and *decrement plan* (3) are divided into two layers: the *commit layer* (4) on the left side and the *plan layer* (5) on the right side.

Commit Layer. In general, the commit layer contains the *commitment* (6) which is made on future events defined by *event types* (8) in the plan layer. One *agent* legally *commits* to the *commitment* (6) which is depicted by a scroll. Agents can either be *inside agents* (7) or *outside agents* (13). A commitment on the decrement plan always has to be in reciprocity with a commitment on the increment plan in order to provide the rational of individual economic activities. In a *contract*, an *inside agent* (7) has to commit to the commitment in the decrement plan and an *outside agent* to the commitment in the increment plan. As for a *schedule*, only *inside agents* can commit to the *commitment*. These restrictions are defined by OCL constraints on the meta model. In the abstract example a commitment `comdec` (6) is made in the decrement plan (3) (upper lane) by the *inside agent* `inside 1` (7a) to execute *event types* (8) specified in the plan layer in the future. In return, a commitment `cominc` (18) is made by the *outside agent* `outside 1`

(13d) in the increment plan (2) (lower lane) to execute *event types* (8) specified in the plan layer in the future. The commitment `cominc` (18) is actually a *commitment series* depicted by a stack of scrolls (18). The special meaning behind a commitment series is, that many commitments (e.g., two orders) can be fulfilled by one event (e.g., a joint delivery).

Plan Layer. The *plan layer* (5) on the right side in general specifies the type of future *events*, their *agents*, and *resources* being involved. These future *events* are defined in the corresponding *duality* of the operational view.

Event Types. The plan layer can contain one-to-many *event types* (8) depicted as dashed hexagons. Contrary to the *operational view* where *events* are specified, in the planning view only *event types* are specified (e.g. regular sale type). The reason for this is, that at the time of planning the events, the actual future events cannot be referred to, because they simply do not exist yet. Thus, only the event type can be referred to. Instead of event types also the sub type *event type series* (12) depicted by a dashed stack of hexagons might be specified in a plan layer. An event type series specifies one-to-many event types of the same kind (e.g. a payment being split up in many partial payments). The event type is related to the *event* in the duality which *fulfills* a commitment (6) or commitment series (18) in the future. A commitment can be fulfilled by one-to-many events and one event can fulfill zero-to-many commitments. This means, that for a commitment, there must be at least one related event in the operational view, but for an event, there does not necessarily exist a commitment. In the abstract example `E1 type` (8a) and `E2 type` (8b) are regular event types and `E3 Type Series` (12) is an event type series. Accordingly, an event `E1` and `E2` as well as an event series `E3 Series` have to exist in the operational view.

Resources and Resource Types. *Event types* are connected to at least one *resource* (9) or *resource type* (10) by *reserve stock-flows*. These resources/resource types refer to resources/resource types in the resource view. If the event type resides on the increment plan (2) the resource will be gained in the future. Otherwise, on the decrement plan (3) the resource will be used in the future when connected by a *reserve use stock-flow* and consumed when connected by a regular *reserve stock-flow*.

A *resource* (9) connected to the event type (also called being reserved) specifies, that the exact identifiable resource is already known at the time of the commitment (e.g. a product with a specific RFID code). Consequently, the exact same resource is also referred to in the future event of the operational view. It is depicted by a regular *solid drop* (cf. 9, `R1`) and references an identifiable resource in the resource view. Similar, a *resource series* (16) has the same meaning for many resources (e.g., 3 books) and is depicted by a *stack of solid drops* (cf. 16, `R5 Series`). It also references an identifiable resource in the resource view.

A *resource type* (10) connected to the event type (also called being specified) can have two specific meanings. Either the referenced resource in the *resource view* is an (i) *identifiable resource* or it is a (ii) *bulk resource*. In the case of an identifiable resource (i), in the future event of the duality a specific resource of this type will be associated. However, at the time of the commitment the exact identifiable resource is not-yet-known (e.g. booking a double bed room in a hotel, but the specific room with number 704 will be assigned at the time of arrival). This resource is depicted by a *dashed drop* (10, `R2 Type`). Similar, a *resource type series* (14) has the same meaning for many resource types (e.g., 10 tons of tuna and 12 tons of carp) and is depicted by a *stack of dashed drops* (14, `R3 Type Series`). In the case of a referenced bulk resource (ii) the actual resource can never be individually identified. Thus, it only defines the quantity of a resource type to be in a stock-flow of a future event (e.g. ordering 1000 liters of heating oil). Similar, a *resource type series* defines the quantity of many resource types to be in a stock-flow of a future event (e.g. ordering 1000 liters of heating oil and 70 liters of diesel).

Agents and Agent Types. *Agents (series)* and *agent types (series)* are reserved/specified for participation in *event types* (8) through *reserve participation* links. Reserved *agents* mean, that the individual agent can already be defined at the time of the commitment and are depicted by a *solid stick figure* (7b, `Inside 3`, 13, `Outside 1`). On the other hand, specified *agent types* mean, that only the type of agent can be defined at the time of the commitment and are depicted by a *dashed stick figure* (11, `Inside 2 Type`); once the commitment is fulfilled by a future event, this *agent type* becomes a concrete known *agent*. An example is specifying that two arbitrary `shop assistants` are needed in the future event, but we do not know the exact individuals yet. An *agent series*/*agent type series* always refers to one-to-many *agent*/*agent types* and is depicted as a *stack of stick figures* (15, `Inside 4 Series`, 17, `Inside 5 Type Series`).

Event types of *schedules* have at least one *inside agent* (7)/*inside agent type* (11) and no *outside agents*. On the other hand, *event types* of *contracts* (1) additionally can have *outside agents* (13). There is usually the same single *outside agent* used for every commitment and event type. However, in some rare cases there might be different *outside agents* as well as even *outside agent types*.

5 REA-DSL Evaluation

The development of REA has been following the *design science* approach in IS by Hevner [10]. In our work the designed artifact is the REA-DSL comprising a well-defined meta model and an appropriate graphical notation. For the design of the REA-DSL we followed the methodological steps for designing domain specific languages suggested by Strembeck and Zdun [11]: Accordingly, we started with (1) the identification of elements in the REA ontology. Next, we (2) derived the abstract syntax of the REA model including the core language model and the language model constraints and (3) defined the DSL behavior, i.e. determining how the language elements of the DSL interact to produce the intended behavior. Once we had reached a stable state, we defined the DSL concrete syntax (4). Finally, we implemented a modeling tool support for the DSL.

With respect to evaluation we (i) demonstrated the technical feasibility by means of a tool implementation, (ii) performed a functional test based on existing REA models, and (iii) conducted interviews with experts. The tool is based on the

Microsoft Visual Studio 2010 Visualization & Modeling SDK. The REA-DSL tool incorporates the REA-DSL meta model as presented in this paper and in our previous paper [5]. In addition, custom code enables to apply additional constraints to the REA-DSL models to adhere to the REA rules.

In the functional test we demonstrated that the REA-DSL is able to address all REA concepts in an appropriate and correct manner. For this purpose we took as input 32 example models which were modeled according to the class-like, original REA representation. Then, we modeled these 32 example models with the REA-DSL by means of our REA-DSL modeling tool. This task helped a lot in the refinement and adjustment of the REA modeling tool. Finally, we were able to describe all 32 models in the REA-DSL.

Eventually, we consulted different experts on our results. Most important we discussed our results with the founder of REA – William McCarthy – on weekly conference calls and incorporated his feedback. We showed them one and the same example in the class-like, original REA representation as well as in the REA-DSL. All the experts agreed that our REA-DSL notation is much more intuitive to describe the underlying business semantics and, thus, may better serve the communication between business experts and IT staff.

6 Summary and Future Work

The Resource-Event-Agent (REA) ontology is a well accepted approach for developing conceptual models for accounting information systems (AIS). The REA concepts are based on well established concepts of the literature in economic theory - which is certainly one of the strengths of REA. However, REA leaves space for diverging interpretations of the relationships between core concepts. Even worse, REA has no dedicated representation format and, thus, no graphical syntax. This is a major shortcoming given its goal to serve as a language for communicating requirements between domain experts and IT staff. Based on these shortcomings we started to develop a domain modeling language for REA that comes with both an unambiguous meta model definition and an intuitive graphical notation. Our first step concentrated on the REA operational layer [5]. This paper extends our previous work by adding concepts of the policy and planning layer to the REA-DSL.

In this paper we proposed extending the domain specific language REA-DSL by commitments and types of the REA policy and planning layers. REA has its roots in the accounting discipline and was extended to a business modeling ontology over the years. Before the REA-DSL was introduced by us in Sonnenberg et al. [5], the REA ontology lacked a dedicated representation built upon a precise formalization. These limitations are tackled by the REA-DSL. However, up to now the REA-DSL just incorporated the core REA concepts resource, event, and agent. The most significant new concepts are commitments, the typification of agents, resources and events, as well as the concept of bulk resources. Commitments define a legally binding obligation to engage in future events. Types can be seen as a grouping of objects with the same properties and can be referenced in commitments if the actual object is not known. Bulk resources refer to resources where the individual resource is not identifiable and traceable. All new concepts have been added to our REA-DSL modeling tool which now covers the operational layer, the planning layer and types of the policy layer.

The ultimate goal of our approach is a business-driven database design or, in other words, the transformation of the REA-DSL artifacts to a conceptual data model for AIS. Accordingly, our future work will add the concepts of attributes and primary keys to resources, events, agents, commitments, etc. Once, this is accomplished transformation rules between the REA-DSL and Entity-Relationship-Diagrams or class diagrams have to be specified. We have already started to work on these issue and first results are very promising. Furthermore, we want to extend the REA-DSL by the concept of terms to model exceptions. Terms specify what happens, if a commitment is not fulfilled and therefore broken. Imagine ordering a fish and it is not delivered at the right time specified in the commitment. In this case terms apply and specify additional events such as penalty payments or additional provided rebates. Once we have completed the work on the conceptual data models and the concept of terms we want to conduct a real world case study by means of the REA-DSL tool.

References

1. Romney, M., Steinbart, P.: Accounting Information Systems. Pearson Education, Limited (2011)
2. McCarthy, W.E.: The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment. The Accounting Review 57(3) (1982)
3. Geerts, G.L., McCarthy, W.E.: An Ontological Analysis of the Economic Primitives of the Extended-REA Enterprise Information Architecture. International Journal of Accounting Information Systems 3(1), 1–16 (2002)
4. Geerts, G.L., McCarthy, W.E.: Policy-Level Specification in REA Enterprise Information Systems. Journal of Information Systems 20(2), 37–63 (2006)
5. Sonnenberg, C., Huemer, C., Hofreiter, B., Mayrhofer, D., Braccini, A.: The REA-DSL: A Domain Specific Modeling Language for Business Models. In: Mouratidis, H., Rolland, C. (eds.) CAiSE 2011. LNCS, vol. 6741, pp. 252–266. Springer, Heidelberg (2011)
6. Geerts, G.L., McCarthy, W.E.: An Accounting Object Infrastructure for Knowledge-Based Enterprise Models. IEEE Intelligent Systems 14(4), 89–94 (1999)
7. Geerts, G.L., McCarthy, W.E.: Modeling Business Enterprises as Value-Added ProcessHierarchies with Resource-Event-Agent Object Templates. In: Business Object Design and Implementation, pp. 94–113. Springer (1997)
8. Geerts, G.L., McCarthy, W.E.: The Ontological Foundations of REA Enterprise Systems, Tulane (March 2005)
9. Gailly, F., Poels, G.: Towards Ontology-Driven Information Systems: Redesign and Formalization of the REA Ontology. In: Abramowicz, W. (ed.) BIS 2007. LNCS, vol. 4439, pp. 245–259. Springer, Heidelberg (2007)
10. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28(1), 75–105 (2004)
11. Strembeck, M., Zdun, U.: An Approach for the Systematic Development of Domain-Specific Languages. Softw., Pract. Exper. 39(15), 1253–1292 (2009)

A Value-Oriented Approach to Business/IT Alignment – Towards Formalizing Purpose in System Engineering

João Pombinho[1,2], David Aveiro[3], and José Tribolet[1,2]

[1] INESC-ID and CODE - Center for Organizational Design & Engineering, INOV, Rua Alves Redol 9, Lisbon, Portugal
[2] Department of Information Systems and Computer Science, Instituto Superior Técnico Technical University of Lisbon, Portugal
[3] Exact Sciences and Engineering Centre, University of Madeira, Funchal, Madeira, Portugal
jpmp@ist.utl.pt, daveiro@uma.pt, jose.tribolet@inesc.pt

Abstract. It is widely recognized that a large percentage of IT initiatives fail from a business perspective. This is attributed to many factors, namely system complexity and change pace. We believe that the system development process itself is a crucial aspect of this state of affairs and a paradigm shift is required. There is a lack a common set of concepts and language to use through an IT development process. Essentially, appropriate models and founded theory for articulating the teleological and ontological perspectives of a system are necessary. In this paper, we present and discuss an innovative value-oriented approach to System Design and Engineering. Our contribution begins by identifying a relevant problem space regarding current approaches, particularly the lack of a sound structure to model a service system's *purpose*. We believe that system modeling with a market mindset will help improving quality and improve change response. The approach draws from a combination of theory based on Enterprise Engineering, Service Science and Value Modeling. A four-layer framework (System, Service, Value and Purpose) is pointed as a conceptual solution for simultaneously representing relevant concerns for promoting dynamic alignment between Business and IT.

Keywords: System Design and Engineering; Enterprise Engineering; Value-orientation; Purpose.

1 Introduction and Motivation

The current global economic crisis context is both an opportunity and an enormous challenge for businesses. Cost reduction through effective reuse, reengineering and innovation being heavily demanded characteristics from enterprises and their supporting systems. Laudon notes that enterprise performance *is optimized when both technology and the organization mutually adjust to one another until a satisfactory fit is obtained* [1]. However, studies indicate as much as 90 percent of organizations fail to succeed in applying their strategies [2]. Misalignments between the *business* and its support systems is frequently appointed as a reason of these failures [1, 3].

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 555–566, 2012.

Aligning Business and IT is a widely known challenge in enterprises. We believe it is mainly due to an essential misalignment: the developer of a system is mostly concerned with its function and construction, while its sponsor is concerned about its purpose, i.e., the system's contribution. Formally integrating the notion of purpose into system development activities requires addressing both the teleological and ontological perspectives in an integrated, bidirectional way [4]. However, Engineering approaches are generally focused solely on the ontological perspective [5].

However, even if this problem was solved, an additional challenge mounts as enterprises are complex systems generally operating in competitive environment. Both their structure of enterprises and the processes that deal with their lifecycle need to balance investments in readiness for change and the effective usage of those investments in terms of value and frequency.

A strong assumption generally hinders solutions: formal organizations are generally created as providers of a *repeatable* and *stable* solution to a given demand. In this context, stable means that there is reasonable belief that the *elements* providing a solution will be continuously available – they are even considered part of the organization. The reason behind this quest for stability is, essentially, the lack of agility in procuring resources on-demand. This leads to compromise between evaluating different solutions to support each business iteration and the time and effort consumed in doing so. It may be argued that the main issue presented can be partially circumvented by using better implementation processes or by increasing modeling coverage and detail. However, we believe its origin is essentially structural. There should be a paradigm shift improving system development with change support in mind. Ideally, a framework should explicitly include the concept of market, with demand/offer dynamics, to address different and innovative solutions to business activity support.

This paper analyzes Business/IT alignment as *system/supporting system* alignment and is structured as follows. Section 2 presents research scoping, with Enterprise Engineering as ground for the Library example that illustrates problem analysis. Then, current challenges are identified and grouped into three problem areas, with a brief related work review. In Section 3 we present a set of core principles and an overview of our Framework. The paper closes with conclusions and contribution summary.

2 Problem Space

2.1 Scope: Systems, Enterprises and Value generation

System has its etymology in the Latin *systema*, *whole compounded of several parts or members*. Skyttner [6] defines it as '*set of elements arranged in such a way that they produce a recognizeable outcome. It is also constituted by a Frontier with the external world, called its Environment*'. The formal definition we will use, from Enterprise Ontology [7], defines the following properties: *composition* – a set of elements of some category; *environment* – a set of elements of the same category, disjoint from the composition; *production* – things produced by elements in the composition and delivered to the environment; and *structure* – a set of influence bonds among elements in the composition, and between them and elements in the environment.

This research addresses *engineered systems*, which are systems deliberately created or changed by means of engineering activities, i.e., with a purpose and rationale. This leaves out of scope naturally occuring geological systems, for instance.

Enterprise Engineering has as a premise that enterprises or, more broadly, organizations, are systems and therefore can be object of systems engineering activities. Organizations have many tipifications, according to their composition and objectives, including: private, public, political, business, educational, healthcare, non-profit, etc. All of these kinds of organizations bring about *value* to their *environment*, either directly or indirectly, so *value* is an unifying concept.

In this paper we present and discuss an innovative, value-oriented approach to System Design and Engineering. It combines work in related domains, such as Enterprise Engineering [7], Service Science [8], and Value Modeling [9].

2.2 Base Theory – Enterprise Engineering and Service Science

Design and Engineering Methodology for Organizations (DEMO)

DEMO [7] is a cross-disciplinary theory for describing the structure and action of organizations, modeled as discrete dynamic systems consisting of social actors. These actors enter to and are responsible for coordinated commitments with each other. Enterprise ontology is a model of an organization in which these commitments serve as models for business transactions. DEMO was chosen because it models the essence of transactions between actors and abstracts away implementation issues.

The *distinction* axiom concerns the separation of knowledge on a specific enterprise in three groups: *ontological (B)*, *infological (I)* and *datalogical (D)*. These are directly related with the abilities *performa* (deciding, judging, etc.), *informa* (deducing, reasoning, computing, etc.) and *forma* (storing, transmitting, etc.). This distinction is very important for distilling the essence of the organization as a social system, with a reported simplification of about 90% in model complexity [10].

In DEMO, an *organization* is defined as a social system, made up of *subjects*, who perform two kinds of *acts*: *production* (P-acts) and *coordination* (C-acts). An *actor* is a subject fulfilling an *actor role*. Actor roles abstract a particular subject performing an act, thus representing the *authority* to perform a particular P-act and related C-acts. *Transactions* are patterns of coordination acts performed in steps: *request*, *promise*, *state* and *accept*. In addition, the cancellation steps of these acts are represented: *decline*, *quit*, *reject* and *stop*. The transaction pattern has the particularity of being able to represent as a path every conceivable transaction. For this reason, it is a good model for the coordination of dual-party interactions between social actors.

To close this brief presentation of DEMO's base theory, we should distinguish two aspects of a system: *teleological*, concerning its function and behavior, a black-box; and *ontological*, about its construction and operation, a white-box [11]. This important distinction, generally absent from other state of the art approaches to our problem, forces both 1) the separation of these concerns and 2) their articulation. The *Generic System Development Process* (GSDP), shown in Fig. 1, addresses the hinge point between teleological and ontological modeling, beginning with the need by a system, the *using system* (US), of a supporting system, the *object system* (OS).

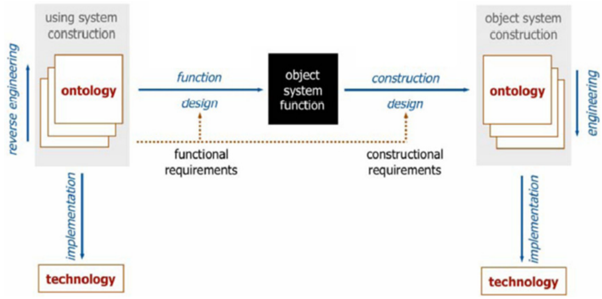


Fig. 1. Generic System Development Process. Adapted from [7]

From the white-box model of the US, one determines the functional requirements for the OS (function design), formulated in terms of the construction and operation of the US. Next, specifications for the construction and operation of the OS are devised, in terms of a white-box model (construction design). The US may also provide constructional (non-functional) requirements. Choices are then made with each transition from the top-level white-box model towards the implementation model.

The GSDP has articulate and clear primitive concepts that reflect the essence of system development. We chose to use it as a reference, since we believe the critical analysis is extensible to other system development processes.

Service Science and Service-Dominant Logic

Both Service Science and Service-Dominant Logic [8] have been around for some years now, rising from the need of a fresh approach to how business is performed. The concept of *service system* [12] is central to both service science (SS) and service-dominant (SD) logic. It is defined as "a configuration of people, technologies, organization and shared information, able to create value to providers, users and other interested entities, through service". Service as a process involves using an actor's resources for the benefit of serving another actor. Social systems are, therefore, of paramount importance in SD-Logic and a conceptual compatibility point with system design and engineering. Additionally, the transaction pattern is common ground between DEMO's transactional pattern and the Service Science concept of Service System, the Interact-Serve-Propose-Agree-Realize model [12]. Analysis of their mapping, presented in [13], allowed concluding that DEMO is quite comprehensive in modeling the flows between these interactions states.

Combining Service Science and Enterprise Engineering is proposed in [13] through the identification of convergence principles. Service Science provides economic theory concerning service exchanges between agents; Enterprise Engineering contributes with the knowledge on designing and engineering those systems.

2.3 Library Example, Issues Identification and Analysis

In order to clarify the problem space, a practical scenario based on the classical DEMO Library case [7] will be used for instantiation.

The Actor Transaction Diagram (ATD) of the Library if presented in Fig. 2. In this example, the elements of the system dealing with the library membership (solid line-bounded area in Fig. 2) are not justifiable as directly bringing value to the customer, who only wants to get hold of a book. However, as it can be seen in Fig. 2 this is all but clear in the ontological (construction) model:

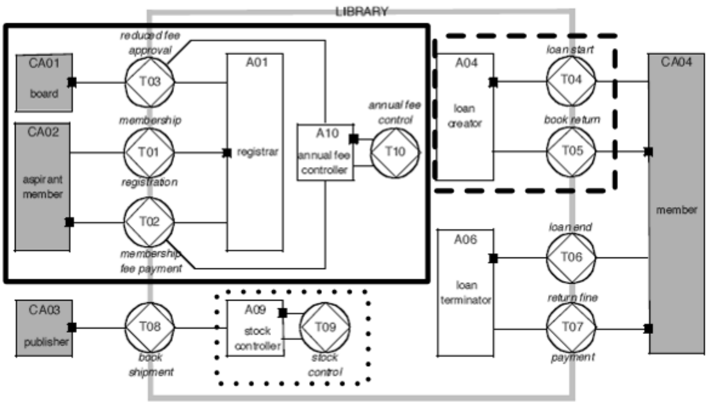


Fig. 2. Library ATD – Construction analysis

Having as a reference the Library's core business, *providing reading content*, some remarks about modeling issues follow: 1) the core service is obscured inside a *loan transaction*, in the area marked by a dashed line; 2) the area bounded by points encloses a support process (stock control) that may need revision, for instance, in a scenario of going digital; 3) inside the solid black line, a *sacrifice* of the customer in obtaining the service and its support (sub)system, let us name it the *Membership Management* subsystem. Considering the later, one must ask if there is really a customer who *wants* a membership? Or was this subsystem included in the Library as a strategy to obtain a fixed amount of income to face, for instance, stocking management? Is this still a problem if the organization does not pay for the books and space? Is it done for profit or simply as a response to the cost of keeping a large library? Is it part of the essential Library business concept, i.e., every library also offers it by definition? Under what conditions its value to the Library should be reviewed?

By analyzing these questions, three main problem areas where isolated:

Defining Value. Value is, by nature, dependent on the stakeholder and, thus, relative. The problems in adequately naming and scoping a service, known in the Service Design community, are symptomatic [14]. Regarding the Library's *purpose*, what is the core transaction for providing value? For instance, should the transaction be named "Loan book" or "Provide (limited-time) access to (reading) content"? Is the "Membership registration" service interesting *per se*, or is it only in the way of getting a book, that is specific to this particular construction of a library? How does a Library compare to a Bookstore or a Publisher, from the customer's perspective? This is why current goal-oriented modeling [15, 16] is not enough: it lacks an independent value structure to refer to. It must be understood that this structure is not subordinate to the service-providing systems, but the other way around, as explained on section 2.4.

Supporting System (de)construction. A system's *construction*, resulting from the development process, is a compiled structure that obscures system/subsystem relations and their motivation. Separating a subsystem from its owner system is hard, especially if it was modeled from a flat description of the operation of the organization instead of an incremental design step. Also, assuming the stability of a value chain is generally unsafe because of change dynamics, which justify the need for a structure where to represent multiple solution scenarios, in order to provide flexibility points instead of a static solution path. Modeling these flexibility points requires the capability of decomposing a system into a chain of value-providing elements.
Modeling system intervention rationale. Implementing a system consists in introducing restrictions on its construction, for instance: 1) assumptions, such as assuming the customer is necessarily a reader; and 2) constraints, such as available technology to offer books, e.g., physical or digital. Still, the rationale of a change is not commonly kept in a reusable way. As an example, regarding the introduction of the Membership subsystem: 1) *What was its purpose? Was it for mitigating the risk of non-return?* 2) *What were the alternatives?* 3) *What were the design principles, constructional principles, assumptions and constraints applied? Do they still hold? Will they hold for any kind of content the library may want to provide, e.g., e-books?* DEMO has been extended [17] to incorporate change dynamics but, currently, still does not model the formal rationale of each change. The GSDP also does not prescribe what to do with the objects supporting intermediate decisions made during the process.

2.4 Related Work – State of the Art Approaches

Enterprise Architecture – Archimate. Archimate is an Enterprise Architecture modeling language with broad practical application. Since becoming a standard of The Open Group, its methodological approach relies on TOGAF [18]. TOGAF has wide audience and results from incorporation of best practices but can hardly be called a formal approach.

In the corresponding architecture framework, three enterprise layers are distinguished: business, application and technology. In comparing with DEMO, the business layer of Archimate and the ontological layer of DEMO are considered. The contents of Archimate's application and technology layers are regarded as implementation and, thus, not directly modeled in DEMO. The business layer of Archimate relates to all three B-I-D layers of DEMO, without clear distinction [19].

While Archimate does not natively address motivation, there are extension proposals [20] with this purpose. *Representing* the motivation has value in itself but, in order to reap the full benefits of addressing the motivation layer, it must go further into *engineering* the business itself. Regarding the connection with business modeling, other approaches combine Archimate and Business Modeling Canvas, a simplified version of the Business Model Ontology [21]. However, none of this approaches uses the concept of *intention*, a characteristic of the so called third wave of approaches, like DEMO [19]. In order to fruitfully combine both approaches, the whole business layer should conform to DEMO, positively constraining the modeling activity. Enforcing the rigorously defined semantics of DEMO into Archimate's business layer is the starting point to perform Archimate-based value modeling.

Value Modeling – e3Value. Value Modeling was selected as it is increasingly recognized that the concept of value assists in improving communication between stakeholders of related systems, particularly Business and IT [22].

e3Value [9] is part of e3family, a set of ontological approaches for modeling networked value constellations. It is directed towards e-commerce and analyses the creation, exchange and consumption of economically valuable objects in a multi-actor network. In e3Value, an Actor is perceived by his or her environment as an economically independent entity, exchanging Value Objects. An enterprise is modeled as an actor in a value network, where the demand and offer market concepts are a natural consequence of the economic context of Value Objects.

Gordjin also introduces the concepts of *value model deconstruction* and *reconstruction* [9], comprising solution reevaluation by analysis and (re)composition of atomic components. Still, no formal decision rationale is supported and no description is provided on how to use the atomic elements beyond the original demand scope.

e3Value has also complementary approaches to modeling strategy [23] and goals [24]. In the first case, the e3forces approach positions of an enterprise in a value web. However, it is insufficient for our objectives as forces are not specified in a way that can be broken down and related with the remainder model constructs. The integration with a goal-oriented approach is very interesting but has the same limitations that GORE approaches, presented next. We acknowledge these approaches and propose to apply e3Value in a way that improves system and subsystem value modeling: inside the boundaries of organizations, as presented in section 3, instead of solely applying to e-commerce relations between formal organizations.

Requirements Engineering – GORE. Goal-oriented Requirements Engineering (GORE) approaches, such as I* and KAOS (for early and late requirements, respectively) were created to trace system (requirements) to its goals, as a way to make purpose tractable. ARMOR [25] is an example of a requirements modeling language focused on motivation, that results from their combination. Despite being an advance in expressing goal structure and tracing, these approaches establish a *goal* as a consequence of having a certain strategy. However, the *goal* must not be mistaken for the strategy itself or the motivation and reasoning behind it.

In order to perform a rational and meaningful analysis it is necessary to make these forces more objective. In the Library example, *Improve Customer Experience*, *Decrease risk of non-return* or *Reduce storage space* are goals that should be decomposed and unambiguously refer to other model constructs, such as with transactions and their production facts, enforcing semantic commitment. Otherwise, they are "simply" the top nodes of a dogmatic hierarchical structure, instead of active elements system engineering. Therefore, we argue that purpose cannot made tractable solely by applying goal-oriented requirements engineering, since known approaches fail to show the real origin of the goal as a result of upstream relations [4]. As we will see in the following section, specifying motivation in term of the needs of another system in successive engineering processes brings a greater level of objectiveness.

3 Towards a Solution: Principles and Framework Overview

3.1 Principles of a Different Way of Thinking and Modeling

From the analysis presented in the last section, we conclude a new fundamental approach is necessary. The mindset used to devise it is using current state of the art approaches for localized modeling, with an improved teleological drive in terms of methodology and principles, presented in the remainder of this section.

Business Service Engineering. In most approaches, the components of the business layer can be modeled without restrictions, with the resulting incompleteness and incoherencies. For instance, Archimate's definition of Business Services lacks DEMO's B-I-D distinction. Its incorporation into Archimate is of utmost relevance to our research, since it allows 1) separating B-I-D in a layered manner and 2) raising the issue of modeling the relations between B-organizations. Therefore, we propose Business Services are validated with DEMO semantics, both in terms of completeness, by matching with the Transactional Pattern, and isolating essential (ontological – B) transactions. In turn, DEMO should relate to an e3Value model by matching actors and transactions. DEMO and e3Value have been related in [26] but not ontologically matched. A matching ontology between both is presented and discussed in [4, 27], introducing, for instance, the concepts of value and economic reciprocity to DEMO transactions and the full transactional pattern to e3Value. This allows transforming an e3Value Value Exchange into DEMO's system construction primitives.

We note that, during system development, we use e3Value without enforcing *economic independence* between actors. This restriction should only be applied in final scenarios, not for design time and exploring innovative solutions as it may leave out important business model features. For instance, in sourced activities may be taken for granted and, thus not modeled as economic independent. More importantly, transactions may not be identified at all – take for instance, *book counseling* in the Library example, which ends up being a differentiator from other offers on the market.

Define *purpose* through Recursive System Engineering. Modeling purpose as the contribution of a given system's production to its environment, as described in [4] is a very powerful teleological concept. It is important to note that the concept of *purpose* is higher in abstraction than the one of *goal* and that the later is subordinate to the former. However, *purpose* is frequently dismissed as a strategic concern and we are not aware of formal, structured, definitions of purpose that can be used as input to a system development process, neither traced back from a developed system. In this framework, purpose is as much of a strategic concern for a given system as it is a need by another. Any given complex system can be decomposed into more granular systems chained together: the rationale for forming each link is the same that should exist between the elements of a system as the same concepts will recursively apply. Therefore, the *purpose* of a system (OS) must be defined in terms of the value it provides to other formal system(s) (US). In turn, this formal system will assume the role of OS towards another US.

Model System Development Rationale. Building on the previous principles, there is now sufficient information and conceptual tools to model each Demand (US)/Offer (OS) pair in the chain and, if relevant, to zoom in into the OS and initiate modeling of its elements reified as independent systems. For each instance of the GSDP, the application of principles and requirements (both function and construction), driven by the needs of the US, progressively filter candidate OS's in the solution market. These primitives are, indeed, the filtering criteria that can now be specified 1) in terms of value, using a common referential and 2) relatively to adjacent elements in the value chain. This enables reasoning about the development process steps and systematically reevaluating previous assumptions and decisions.

3.2 Framework Overview

Our framework is based on the concept of a solution market as a means to provide purpose orientation. Systems provide services, which the customer (*using system*) gets from the provider (*object system*). Services are then valued in a market context and may be used in multiple solution chains. This way, the integration of teleological and ontological visions of a system is done by assigning market-based value semantics to each system construction element. Modeling the intermediate steps between the formulation of a need and its fulfilling by a system is, therefore, indispensable for informed and rational decisions.

In the Library example introduced in section 2, the system may be used by a customer to solve an *information need* problem, but it can also solve a *gifting* problem. According to the demand segments the system's owner wishes to address, and competing offers, different system design and engineering decisions are made. Modeling these teleological aspects is not trivial and is commonly regarded as subjective. However, by distinguishing the *contribution*, *function* and *construction* perspectives [5] business service specification can be improved.

Analyzing the Library's *loan start* transaction, we identify the value object that is functionally offered by the Library in exchange for a membership fee, not as the *book* but as its *temporary possession*. This may or not serve the interests of *using systems*, they are bound to returning the book. By defining the value of the service offered by the Library to a specific *using system* in a market context, it is possible to compare alternative solutions to procure inputs for its own value activities, i.e., *reading* or *gifting*. The *contribution perspective* abstracts any implementation choices or provisioning mechanisms. Hence, it brings the Library's production to an essential level, the first step in allowing comparison to other alternatives of bringing about such *production fact*. Some examples are internet ordering, borrowing from a friend, acquiring a digital version or even downloading an illegal copy. Each alternative introduces an offer at the solution market with specific value and dependencies, which end up providing different end-user experiences.

The combination of approaches, including DEMO, Service Science and e3Value was achieved first by theory analysis and combination, and then by ontological mapping, with modeling restrictions introduced from each theory towards the others. A formal mapping between DEMO and e3Value [9] allows enforcing the relations between perspectives: *construction* supports *function* and *function* supports *contribution*.

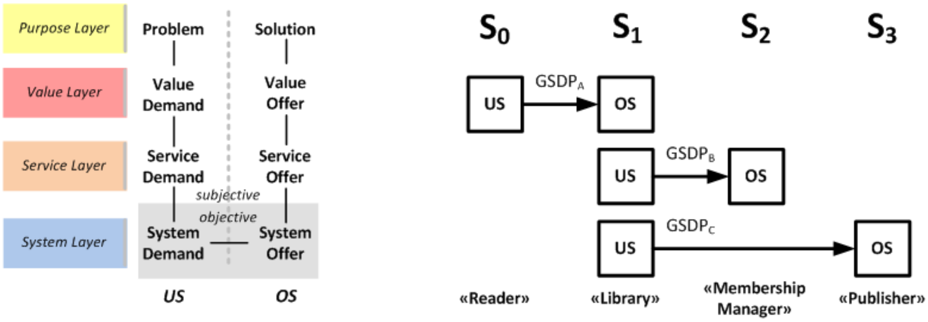


Fig. 3. Framework Overview (left) and recursive GSDP application (right)

Drawing from the principles presented in the previous section, we propose simultaneously modeling a set of layers for a given system: System, Service, Value (Market) and Purpose (Problem Solving). Their relative positioning is represented on the left part of Fig. 3. A brief description of each layer follows:

- The *System* layer defines how a service is assembled from individual elements and provided by a concrete system. Both *construction* and *function* perspectives are defined at this layer. The solution provided by a system is only effectively made operational two concrete systems, assuming the demand (US) and offer (OS) roles, connect. This concept is represented by the *objective* box in Fig. 3.
- The *Service* layer models the demands/offers a system makes/receives to/from its environment. It encapsulates a *system* as a *service system*, by providing a *partial* black-box model, framed with contract and operation conditions. Service differs from function since it incorporates the concept of *willingness* to engage into transactions. This concept is represented by a Value Port (e3Value), welcoming service interactions, as defined SD-Logic's ISPAR Model [12]. For instance, a Library could functionally provide book selling services to customers but chooses not to.
- The *Value* layer defines value exchanges composed of service offers and demands. For instance, it allows comparing a Bookstore and a Library as solutions to *provide reading content*: one fully transfers value object (*book*) rights and has unitary costs, while the other only offers temporary access at a flat rate. This layer is critical in relating the services provided by a system and a specific stakeholder, as value is always uniquely and phenomenological determined by its beneficiary [8].
- The *Purpose* layer defines the contribution of an *object system* as a solution provider to the demands of specific *using systems*. It connects service systems based on value object chaining, e.g. the Result Chain presented in [28] and, in a more narrow but formal sense, to DEMO's Result Structure [7]. For instance, the Publisher contributes by providing *book copies* which the Library can, in turn, loan to readers. Together with *service* and *value*, *purpose* fully addresses the *contribution perspective* as defined in [5] by re-scoping the development process to adjacent nodes on the value chain and recursively defining system relations (Fig. 3, right).

4 Conclusion

We presented a proposal for defining a system's purpose, offering a multi-disciplinary approach that integrates contributions from multiple relevant areas, resulting in a diverse conceptual foundation. Its most differentiating feature is in the way of thinking, by finding new principles that allow conceiving the relation of a system with its elements as if they are in an open market. This essentially means modeling them as separate actors from an economic viewpoint, which we argue improves modeling completeness and quality. Also, focusing on value as the bottom line means having a traceable relation between every element of the model and purpose, which assists in communication between stakeholders.

Our contribution is composed by: 1) the identification of a relevant problem space in current approaches, particularly the lack of a sound structure to model and provide an ongoing referential for *purpose*; and 2) the definition of a *conceptual framework* that addresses the issues identified in section 2 of this paper. It integrates the core concepts and their relative positioning in a layered manner, distinguishing the concepts that define a problem/solution pair end-to-end, from need to implementation.

We believe performing system modeling with a market mindset will help improving quality and change response. In this paper, we presented a combination of Enterprise Engineering, Service Science and Value Modeling. *Business* and its supporting systems can only be effectively aligned by using a sound conceptual theory, together with the corresponding methodologies and applications, integrating both the *contribution* (why), *function* (what) and *construction* (how) dimensions of a system. Formally addressing the contribution perspective of a system is critical to trace system construction to its purpose and improving Business/IT alignment.

References

1. Laudon, K.C., Laudon, J.P.: Management Information Systems: Managing the Digital Firm. Prentice Hall
2. Kaplan, R.S., Norton, D.P.: Strategy Maps: Converting Intangible Assets into Tangible Outcomes. Harvard Business School Press, Boston (2004)
3. Henderson, J.C., Venkatraman, N.: Strategic alignment: leveraging information technology for transforming organizations, vol. 32(1), pp. 4–16 (1993)
4. Pombinho, J., Aveiro, D., Tribolet, J.: Towards Objective Business Modeling in Enterprise Engineering – Defining Function, Value and Purpose. In: Albani, A., Aveiro, D., Barjis, J. (eds.) EEWC 2012. LNBIP, vol. 110, pp. 93–107. Springer, Heidelberg (2012)
5. Op 't Land, M., Pombinho, J.: Strengthening the Foundations Underlying the Enterprise Engineering Manifesto. In: 2nd Enterprise Engineering Working Conference. Springer, Delft (2012) (forthcoming)
6. Skyttner, L.: General Systems Theory: Problems, Perspectives, Practice, 2nd edn. World Scientific Publishing Co. Pte. Ltd., Singapore (2005)
7. Dietz, J.L.G.: Enterprise Ontology: Theory and Methodology. Springer (2006)
8. Vargo, S.L., Lusch, R.F., Akaka, M.A.: Advancing Service Science with Service-Dominant Logic: Clarifications and Conceptual Development. In: Handbook of Service Science. Springer (2010)

9. Gordijn, J.: Value-based requirements Engineering: Exploring innovatie e-commerce ideas. Vrije Universiteit Amsterdam, Amsterdam (2002)
10. Dietz, J.L.G.: The Atoms, Molecules and Fibers of Organizations. Journal Data & Knowledge Engineering - Special Issue: The Language/Action Perspective Archive 47(3) (2003)
11. Dietz, J.L.G.: Architecture - Building strategy into design. Netherlands Architecture Forum, Academic Service - SDU, The Hague, The Netherlands (2008)
12. Maglio, P.P., et al.: The service system is the basic abstraction of the service science. Information Systems and E Business Management (2009)
13. Pombinho, J., Tribolet, J.: Service System Design and Engineering – A value-oriented approach based on DEMO. In: 3rd International Conference on Exploring Service Science, Geneva (2012)
14. Bell, M.: Service-Oriented Modeling: Service Analysis, Design and Architecture. John Wiley & Sons, New Jersey (2008)
15. Lamsweerde, A.V.: Goal-Oriented Requirements Engineering: A Guided Tour. In: International Symposium on Requirements Engineering, Toronto (2001)
16. Regev, G., Wegmann, A.: Where do Goals Come from: the Underlying Principles of Goal-Oriented Requirements Engineering. In: 13th IEEE International Requirements Engineering Conference (RE 2005). IEEE, Paris (2005)
17. Aveiro, D.: G.O.D (Generation, Operationalization & Discontinuation) and Control (sub)organizations: a DEMO-based approach for continuous real-time management of organizational change caused by exceptions. UTL, Lisboa (2010)
18. Lankhorst, M., van Drunen, H.: Enterprise Architecture Development and Modelling - Combining TOGAF and ArchiMate (2007)
19. Ettema, R., Dietz, J.L.G.: ArchiMate and DEMO – Mates to Date? In: Albani, A., Barjis, J., Dietz, J.L.G. (eds.) CIAO!/EOMAS 2009. LNBIP, vol. 34, pp. 172–186. Springer, Heidelberg (2009)
20. Engelsman, W., et al.: Extending enterprise architecture modeling with business goals and requirements. Enterprise Information Systems 5, 9–36 (2011)
21. Osterwalder, A.: The Business Model Ontology - a proposition in a design science approach. Universite de Lausanne (2004)
22. Cameron, B., Leaver, S., Worthington, B.: Value-Based Communication Boosts Business' Perception of IT. Forrester Research (2009)
23. Pijpers, V., Gordijn, J.: e3forces: Understanding Strategies of Networked e3value Constellations by Analyzing Environmental Forces. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007 and WES 2007. LNCS, vol. 4495, pp. 188–202. Springer, Heidelberg (2007)
24. Gordijn, J., Petit, M., Wieringa, R.: Understanding Business Strategies of Networked Value Constellations Using Goal and Value Modeling. In: Proceedings of the 14th IEEE International Requirements Engineering Conference. IEEE Computer Society (2006)
25. Quartel, D., et al.: A goal-oriented requirements modelling language for enterprise architecture. In: IEEE International Enterprise Distributed Object Computing Conference, Auckland, New Zealand (2009)
26. Weigand, H., Heuvel, W.-J.V.D.: A Conceptual Architecture for Pragmatic Web Services. In: First International Conference on the Pragmatic Web, Stuttgart, Germany (2006)
27. Pombinho, J., Tribolet, J.: Modeling the Value of a System's Production – Matching DEMO and e3Value. In: 6th International Workshop on Value Modeling and Business Ontology, Vienna, Austria (2012)
28. Boehm, B.: Value-Based Software Engineering. SIGSOFT Software Engineering Notes 28(2), 4 (2003)

Strategic Alignment Maturity Model (SAMM) in a Cascading Balanced Scorecard (BSC) Environment: Utilization and Challenges

Suchit Ahuja

Purdue University, Department of Computer & Information Technology,
West Lafayette, IN 47906, USA
suchitahuja@purdue.edu

Abstract. SAMM is a useful tool for measuring the maturity of business/IT alignment in an organization at the macro level. However, at the micro level, organizations use several frameworks including cascading BSC, ITIL, COBIT, etc. to align business and IT processes. The complexity of alignment increases with the existence of more than one tier of cascading and usage of different tools or frameworks. Studies have shown that measuring business/IT alignment at the micro level is difficult. Therefore, in order to accurately measure outcomes, mapping between metrics at all levels is required. It is also important to establish metrics that are aligned with those prescribed by SAMM. Using a multi-level cascading BSC that was previously published in BUSITAL by this author, this study attempts to apply the underlying components of SAMM and to establish relevant alignment metrics. It also highlights some applicability problems and suggests appropriate solutions for future implementations.

Keywords: Balanced Scorecard, cascading BSC, Strategic Alignment Maturity Model, COBIT, metrics, Key Performance Indicators, Key Goal Indicators, Business Alignment, IT Alignment, aggregated metrics dashboard.

1 Introduction and Background

Organizations constantly strive to maximize use of resources and obtain higher returns on IT investment, by optimizing the integration of IT and business with respect to processes, functions, technologies, systems, and human resources. The traditional view has been that IT is a support function of the business organization. With the advent of globalization and popularity of e-commerce, this paradigm has started shifting significantly. Furthermore, IT is now increasingly integrated with the business functions of the organization. In many sectors, IT has evolved into a strategic differentiator and has transformed into a primary driver of the business. This scenario has provided organizations with an opportunity and an incentive to further integrate and align IT with business. Thus, business/IT alignment is moving towards alignment of IT functions with business functions, in order to enable the organization to derive strategic advantages.

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 567–579, 2012.

Although the benefits of business/IT alignment significantly outweigh the costs, several studies [1], [2], [3] indicate that organizations still lack in the successful implementation of business/IT alignment in terms of both business-to-IT and IT-to-business alignment. The primary reasons for low success rates in business/IT alignment are lack of a uniform definition of business/IT alignment, pursuit of a unilateral strategy for alignment, and lack of an appropriate tool to measure success of business/IT alignment [2]. In order to provide a tool to measure business/IT alignment maturity in an organization, the Strategic Alignment Maturity Model (SAMM) [1] [2] was developed.

In contrast, due to the complexity of aligning business and IT, organizations use a multitude of tools, frameworks and processes in order to best align business and IT functions according to their requirements. As a result, there is an emphasis on the organization-wide implementation of processes for aligning business and IT. Depending on the structure of the organization, such a process may consist of one or more tiers. Balanced Scorecard (BSC) [1] is a widely used tool for business/IT alignment purposes. However, using a standalone BSC for business/IT alignment is almost never sufficient to achieve success with business/IT alignment. BSC can be used at a strategic level in order to align business and IT, but it is also used in conjunction with tactical frameworks such as COBIT and with operational frameworks such as ITIL [17]. Therefore, organizations choose to use a combination of tools or frameworks in order to achieve business/IT alignment success [18]. A complex cascade of key performance indicators (KPIs), key goal indicators (KGIs) and other metrics are used in order to ensure traceability of strategic alignment throughout the organization [17][18][19].

These metrics may not necessarily represent the set of metrics proposed by SAMM. Studies have shown that in order to use SAMM effectively, native metrics and tools must be tweaked in order to fit criteria established by SAMM [4]. This paper aims to highlight this issue with respect to usability of SAMM. While SAMM can be used to measure business/IT alignment maturity for an organization, it is a highly subjective tool and most studies show the use of a survey instrument to derive overall maturity. Interestingly, if the organization uses different levels of cascades in strategic, tactical and operational alignment, at the micro level, the reported results of the surveys may vary with the specific tier at which the data is collected. Furthermore, if different tiers of an organization report different data with regards to business/IT alignment maturity, it would be difficult to derive an overall organizational business/IT maturity level. A very similar problem has been highlighted by a pervious study in the context of aggregation of organizational metrics while using a cascading BSC [8]. The study also points to the fact that measuring maturity may involve distinct frameworks with each prescribing its own maturity model. Thus, it is important to create mappings between maturity models being used for different frameworks in order to derive an overall organizational maturity model with respect to business/IT alignment [9].

The following sections will elaborate on how SAMM provides a unified and aggregate maturity model to measure business/IT alignment success and maturity, but inherently lacks in prescribing steps for simplifying the measurement process and guidelines for implementation in varied environments. It is important to note that this study is an attempt at understanding the complexities of measuring business/IT alignment in a varied environment. The long term goal is to find an objective and comprehensive solution to the problem of measuring business/IT alignment maturity at the micro level and such a scenario is subject to various constraints including the environment in which alignment is being measured. For this paper, a previously published sample cascading BSC environment is used.

2 Analysis Approach

This paper provides an analysis of SAMM at micro levels in a cascading BSC environment. Individual components (SAMM and cascading BSC) will be analyzed with a focus on two aspects: 1) Alignment measurement parameters native to BSC and 2) Maturity model used in a cascading environment and its relevance to SAMM. The goal is to investigate if the cascading BSC inherently provides information that can be used in conjunction with SAMM to derive a maturity level of business/IT alignment (with respect to criteria prescribed by SAMM). It is noteworthy that the goal is not to provide a working model of a combination of both the BSC and SAMM, but to provide insights into the problems involved in this process and the required steps in order to mitigate these problems.

2.1 Relevance to Previous Research

The author has published an example of a cascading BSC previously [9]. This framework is an integrated strategic framework for Information Security Management and uses COBIT for governance and Security Engineering Capability Maturity Model (SSE-CMM) for maturity measurement. The cascading BSC used for strategic alignment is displayed in Figure 1 and the maturity model used is shown in Figure 2. These two components are used as major building blocks for analysis where cascading BSC forms the micro-level environment where alignment maturity needs to be measured and the SSE-CMM maturity model shows how strategic alignment is measured using mappings between existing models due to lack of a comprehensive mechanism.

3 Components

In this section, the two major components that form the basis of investigation for this paper – Strategic Alignment Maturity Model (SAMM) and the cascading Balanced Scorecard (BSC) – are outlined and their functions are explained from the perspective of micro-level strategic alignment.

3.1 Strategic Alignment Maturity Model (SAMM)

SAMM is a tool that can be used to measure alignment maturity [3]. SAMM proposes that IT-Business alignment can be captured according to six areas of maturity. Table 1 shows the areas and levels of maturity. It also shows some sample attributes of each area that are relevant to the measurement.

SAMM prescribes the following areas in order to measure organizational maturity: A) Communication; B) Competency / Value measurement; C) Governance; D) Partnership; E) Scope and architecture; and F) Skills.

For each of these areas, this maturity model classifies the alignment between business and IT into five levels: 1) Initial / Ad hoc process; 2) Committed process; 3) Established / Focused process; 4) Improved / Managed process; 5) Optimized process.

Table 1. SAMM and its areas, maturity levels and attributes [3].

Area	Level	Attribute
Communication Liaison Effectiveness, Understanding of Business by IT, Understanding of IT Inter/Intra-organizational Learning/Education, Protocol Rigidity, Knowledge Sharing	1	Business/IT lack understanding
	2	limited business/IT understanding
	3	good understanding; relaxed communications emerging
	4	Bonding, unified
	5	informal, pervasive
IT Value IT Metrics, Business Metrics, Balanced Metrics, Service Level Agreements, Benchmarking, Formal Assessments/Reviews, Continuous Improvement	1	some technical measurements
	2	measures functional cost efficiency
	3	measures some cost effectiveness; dashboard established
	4	measures cost effectiveness; some partner value; dashboard managed
	5	measures extended to external partners
IT Governance Business Strategic Planning, IT Strategic Planning, Budgetary Control, Steering Committee(s), Prioritization Process	1	no formal process, cost center, reactive priorities
	2	Tactical at functional level, occasionally responsive
	3	relevant process across the organization
	4	managed across the organization
	5	integrated across the firm and partners
Partnership Business Perception of IT Value, Role of IT in Strategic Business Planning, Shared Goals, Risk, Rewards/Penalties, IT Program Management, Relationship/Trust Style, Business Sponsor/Champion	1	conflict; IT is a cost of doing business
	2	IT emerging as an asset; process enabler
	3	IT is as an asset; process driver; conflict seen as creative
	4	IT enables/drives business strategy
	5	IT-business adaptive and improvise together
Scope & Architecture Traditional, Enabler/Driver, External, Standards Articulation, Architectural Integration, Architectural Transparency, Agility, Flexibility, Manage Emerging Technology	1	Traditional (e.g., accounting, email)
	2	Transactional (e.g., ess, dss)
	3	integrated across the organization
	4	integrated with partners
	5	evolve with partners
Skills Cultural Locus of Power, Change Readiness, Innovation, Entrepreneurship, Management Style, Career Crossover, Training/Education, Hiring and Retaining	1	IT takes risk, little reward; technical training only
	2	differs across functional organizations
	3	emerging value service provider; balanced technical and business hiring
	4	shared risks and rewards
	5	education/careers/rewards across the organization

The above is a general maturity and attribute list prescribed by SAMM and can be tweaked to fit the organizational needs. Several past studies have tried to adapt this maturity model based on survey questionnaires, qualitative responses and quantitative analysis to fit the results within the parameters of SAMM. It was difficult to find studies that highlight difficulties in the micro-level applicability of SAMM with respect to the strategic, tactical and operational levels of the organization. Further discussion about the differences between such past studies and the approach of this paper is presented in the analysis section.

3.2 Cascading BSC Approach and Alignment Maturity Model

Balanced Scorecard [10] by definition is a performance management system that enables businesses, business units and functional business areas to drive strategies based on goal definitions, measurement and follow-up [11]. According to the Balanced Scorecard Institute [12], "cascading a balanced scorecard means to translate the corporate-wide scorecard (referred to as Tier 1) down to first business units, support units or departments (Tier 2) and then teams or individuals (Tier 3). The result should be to focus across all levels of the organization. The organizational alignment should be clearly visible through strategy, using strategy maps, performance measures and targets, and initiatives. Scorecards are used to improve accountability through objective and performance measure ownership, and desired employee behaviors are incentivized with recognition and rewards." This cascading process can help achieve a better fit for the organization and provide a customized scorecard that can produce improved results. The cascading balanced scorecard approach (between business and IT) can be successfully used as a strategic management tool [13], [14], [15], and [16].

There are two aspects that need to be considered within the BSC context:

1) *The cascading BSC model and metrics related to it*: In Fig. 1, a cascading BSC is displayed for strategic alignment of business, IT and information security functions within an organization [9]. Output objectives and their respective metrics from the first tier serve as input to the next tier and an overall organizational set of metrics is derived in order to measure success of the initiatives at each tier. It must be noted that there is no comprehensive set of metrics to track the success of alignment. This is a fundamental problem that needs to be addressed.

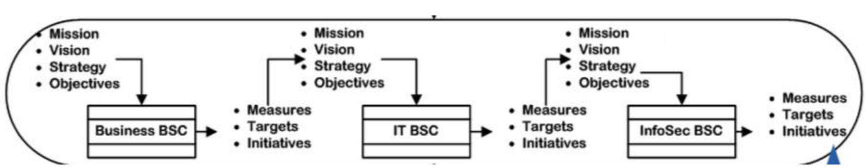


Fig. 1. An integrated framework for Business-IT-Information Security Alignment using cascading BSC approach [9]

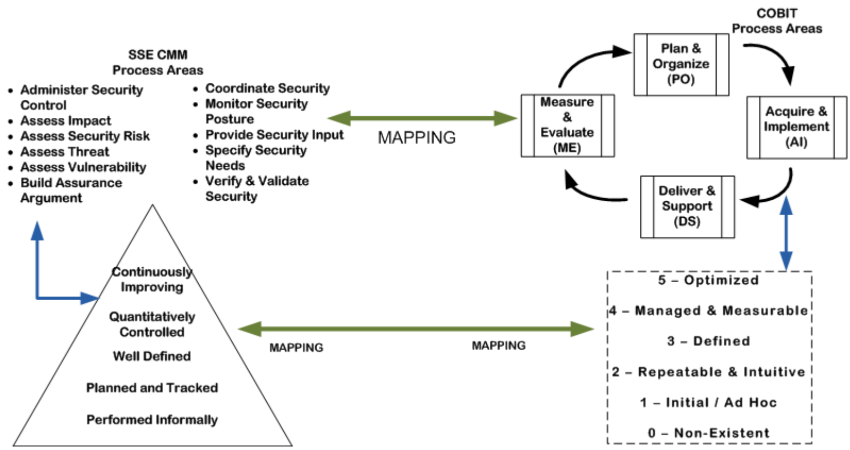


Fig. 2. A maturity model used for measuring alignment maturity using process mapping between COBIT process areas and SSE-CMM [9]

2) *The maturity model used to measure overall maturity of the organization*: Fig. 2 provides an example of measuring alignment maturity in a cascading BSC environment with use of different maturity models by creating a process mapping between them and was published in [9]. Since the context of research in [9] was strategic information security management, the maturity model prescribed by COBIT is used by creating process mappings with Systems Security Engineering Capability Maturity Model (SSE-CMM). It shows how SSE-CMM process areas can be mapped with COBIT domains and the corresponding maturity models can be aligned, in order to effectively measure strategic alignment of information security processes.

This is done primarily due to lack of a tool that can provide a comprehensive maturity model and metrics. Although SAMM can be used to mitigate this gap, it may not provide a solution that takes into account factors such as variation in the definition of quantitative metrics when varied frameworks are used for alignment. For example, if the previous study was about alignment of IT service strategy and the ITIL framework was used instead of COBIT, then Capability Maturity Model Integration (CMMI) would logically provide a better fit in terms of maturity model than SSE-CMM. Thus, depending on organizational requirements the maturity model used may vary and it is difficult to measure alignment using a standard set of criteria in such diverse and varied environments. In order to ease the process of measurement, some fundamental metrics may have to be introduced at every tier within the cascading BSC, such that deriving a clear maturity level is indeed possible.

4 Analysis

This section provides an analysis of problems related to alignment measurement and alignment maturity measurement within a cascading BSC context. The goal is to highlight issues in terms of maturity measurements that currently exist and present some solutions that can be applied.

4.1 Alignment Measurement Problems in a Cascading BSC Approach

Using a cascading BSC approach, several studies [9], [19], [20] have focused on deriving organization-wide Key Performance Indicators (KPIs) and metrics from different tiers within the BSC cascade. These studies highlighted the problem of existence of numerous metrics at different tiers without guidance about how to integrate or aggregate such metrics. In [20] a cascade of KPIs and Key Goal Indicators (KGIs) was used in order to mitigate the problem of integration and aggregation of metrics and [9] showed how this approach can be used to establish traceability with KPIs and KGIs within the organization.

Firstly, such KPIs and KGIs (along with other metrics) are relevant to strategic alignment and are excellent in terms of aligning strategy and tracing process performance within the organization. These metrics may not be suitable to measure alignment success or alignment maturity. When organizations design BSCs, measurement of alignment success is not taken into consideration, as the primary focus is only on strategic alignment via use of objectives and indicators. Secondly, at each tier different metrics are used and therefore result in different alignment measurement criteria for each tier. Without guidance about measuring comprehensive strategic alignment maturity, it is difficult to achieve quantifiable results.

In order to highlight some of the difficulties involved with respect to use of SAMM at the micro-level, Table 2 shows the InfoSec BSC from [9]. This is the final tier of the cascading BSC approach in an information security strategy alignment project and mitigates the gap between business objectives and IT objectives via use of a specific BSC for information security strategy. The goal of using a specific InfoSec BSC is to provide clear top-down and bottom-up traceability in terms of objectives, metrics, targets and initiatives. Thus in Table 2, “S1-O1-M1-T1” represents the corresponding strategic initiative, objective, metric and target established at the business and IT levels. The lower part of the table represents initiatives, metrics and targets specific to the information security business unit. Importantly, the objectives and metrics of the InfoSec BSC are specific to information security strategy alignment, without necessarily providing any metrics to measure maturity of alignment. Such alignment metrics are also missing from the Business BSC and IT BSC used in [9]. Thus, using an InfoSec BSC would ease the process of strategic alignment based on metrics and/or KPIs and KGIs, but measuring maturity of overall alignment within the business unit or the organization as a whole would prove to be difficult.

To further highlight the problem discussed above, Fig. 3 displays the missing “alignment metrics” that can be used as guidance for tracking business/IT/infosec alignment maturity in a cascading environment. In the absence of such alignment maturity metrics, which in this example have been derived from [3], the output from each tier of the cascading BSC would only be useful as a strategic metric and not as an alignment maturity metric.

Table 2. InfoSec BSC showing traceable metrics for strategic alignment to Business & IT [9]

Objective	Objective Detail	Target ID	Measurement Details	Target
Strategy: **S1** Objective: **O1** *Secure CPOE Integration*	Monitor & Report	S1-O1-M1-T1	% of security CPOE events generated per day vs. total CPOE events	< 10%
	Problem Tracking	S1-O2-M1-T1	% of reported security issues traced vs. unresolved	90%
	Violations	S1-O3-M1-T1	% of security violations detected per day	100%

Initiative	Measurement Details	Target	Initiatives
S1-O1-M1-T1-I1 *Objective: Secure CPOE Integration*	% of security CPOE events generated per day vs. total CPOE events	< 10%	+ Enhance CPOE security evaluation process + Increasing physician awareness by providing additional training + Increasing application awareness by providing additional training to configuration mgmt. teams
S1-O2-M1-T1-I1	% of reported security issues traced vs. unresolved	90%	Historical tracking tools, training for current staff, ticketing and reporting system
S1-O3-M1-T1-I1	% security violations detected per day	100%	Installation of IDS / IPS

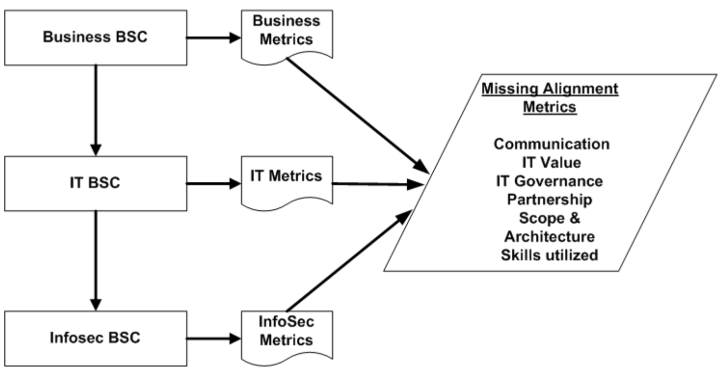


Fig. 3. Missing alignment metrics in a cascading BSC environment

Hence, it is important to distinguish between metrics that are used for strategic alignment and metrics that are used to measure maturity of strategic alignment.

4.1.1 Mitigation of Alignment Measurement Problems in a Cascading BSC Approach

As discussed above, a mechanism must be devised to mitigate the lack of "alignment measurement" and "alignment maturity measurement" metrics at each tier in a cascading BSC. As a first step, it is important to recognize that alignment must not be measured at each individual tier in the cascade, but at the intersection of two tiers, thus staying true to the very definition of alignment. Fig. 4 shows how alignment metrics based on SAMM can be used between two tiers of a cascading BSC in order to derive a common "alignment metric document" that can serve as a baseline for deriving strategic alignment maturity. To that end, the organization must devise alignment metrics based on its own requirements and taking into account its current metric collection mechanisms, complexity of enterprise architecture, domain of business, governance model, etc. Furthermore, it is important to note that this is not a comprehensive solution to measuring strategic alignment maturity. This is a conceptual step that shows how organizations can move towards adoption of SAMM and other strategic alignment maturity models, even while functioning in a complex alignment environment that is not natively designed to support initiatives like SAMM.

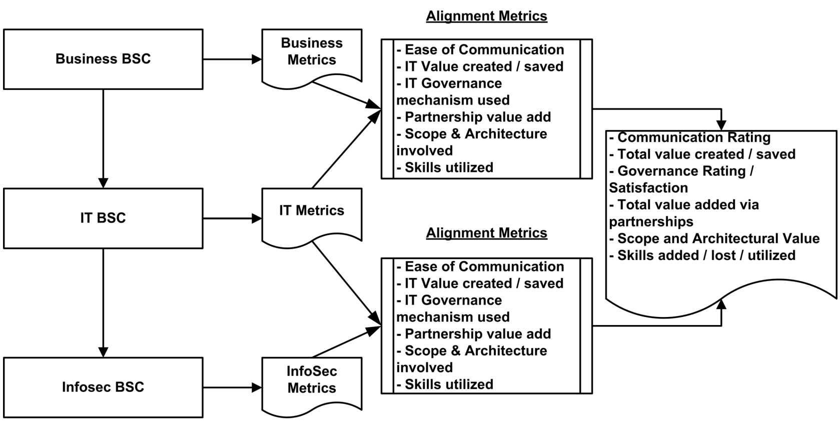


Fig. 4. Use of "alignment metrics" between each cascade/tier in BSC, in addition to using traditional metrics

4.2 Maturity Measurement Problems in a Cascading BSC Approach

Problems that exist in using a maturity model to assess strategic alignment maturity are well documented [3], [4], [5], and [6]. Several studies have attempted to solve this problem by using well-established industry maturity models like CMMI, SSE-CMM, COBIT Maturity Model, etc. [8], [9], [17]. These studies have mostly used a process mapping approach to create process maps [21] between the maturity model and other components within their respective frameworks. Problems related to measurement of maturity in such frameworks are also well documented. These problems include:

1. Lack of a comprehensive maturity model to measure overall maturity of the alignment process [8], [9].
2. A mismatch between maturity models proposed by different elements of the tools/frameworks used in the alignment process [8], [9]

Some recent studies have cited SAMM as a solution. Most of these studies are qualitative and based on surveys, thus covering only macro-level aspects of measuring alignment maturity [3], [4], [22], [23], [24]. The survey and questionnaire tools are excellent in terms of providing a perception of strategic alignment of the organization via the person responding to the survey. It would be interesting to investigate if these responses remain the same or vary with respondents from the same organization but different business units. Furthermore, there is no evidence about the tools or frameworks used by the surveyed organizations to measure strategic alignment maturity.

Consequently, it is very important to try to apply the principles of SAMM at the micro level, in order to provide uniformity in alignment maturity at all levels of the organization – strategic, tactical and operational. Organizations that have different business units reporting different levels of maturity in terms of strategic alignment, often pose a problem of measuring overall organizational strategic maturity. Fig. 5 displays a scenario where different business units may report varying maturity levels with respect to SAMM criteria, thereby making it difficult to calculate overall strategic alignment maturity of the organization.

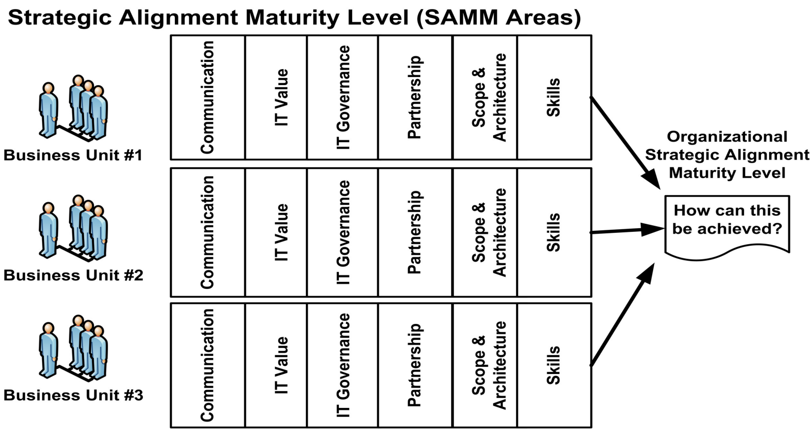


Fig. 5. SAMM areas being reported at different maturity levels by different business units

Therefore, it is important to establish and introduce metrics related to SAMM areas into alignment mechanisms already in use. Further investigation into how this can be achieved is necessary and forms the basis of the future work of this study.

5 Conclusion and Future Work

SAMM is an excellent model to measure maturity of strategic alignment of an organization at the macro level. However, at the micro level, challenges exist in terms of using metrics derived from traditional alignment frameworks such as cascading BSC. The metrics from cascading BSC framework can be easily used to align business and IT strategies, but these metrics are not inherently designed to track strategic alignment maturity.

This paper has highlighted the following aspects with respect to alignment metrics:

1. The need to establish valid metrics, based on attributes and areas prescribed by SAMM, in order to ensure that strategic alignment maturity can be objectively measured and tracked.
2. An analysis of problems underlying application of SAMM at the micro level.

It provides a conceptual solution with respect to establishing “strategic alignment maturity” metrics. Nevertheless, there is still a problem in connecting specific aspects of existing and widely-used maturity models with those prescribed by SAMM which needs to be addressed in detail.

This paper is not a comprehensive solution to the problems mentioned above. It provides guidance to ease the application of SAMM in complex environments such as those with cascading BSCs. Thus, future work involves:

1. A step-by-step application of metrics that can help track and measure strategic alignment maturity in a cascading BSC environment.
2. Pre-define domain-specific metrics with respect to SAMM attributes in addition to the six areas already covered.
3. Attempt to provide objective metrics, such that an auditable and traceable maturity model can be established, while applying underlying principles of continuous improvement that can help sustain the established processes.

The above points provide a basis for future studies with goals that would help achieve seamless integration of SAMM with micro-level alignment frameworks.

References

1. Kaplan, R.S., Norton, D.P.: The balanced scorecard–measures that drive performance. Harvard Business Review 70(1), 71–79 (1992), doi:10.2307/1323539
2. Walter, B.: Aligning information systems with business strategy. The Journal of Strategic Information Systems 1(4), 205–213 (1992), doi:10.1016/0963-8687(92)90036-V
3. Luftman, J., Kempaiah, R.: An Update on Business-IT Alignment: “A Line” Has Been Drawn. MIS Quarterly Executive 6(3), 165–177 (2007)
4. Khaiata, M., Zualkernan, I.A.: A Simple Instrument to Measure IT-Business Alignment Maturity. Information Systems Management 26(2), 138–152 (2009), doi:10.1080/10580530902797524
5. Data Driven Reporting and Communication about IT | IT Policy Compliance (n.d.), `http://www.itpolicycompliance.com/blog/data-driven-reporting-and-communication-about-it-2/` (retrieved February 21, 2012)
6. Information Systems and the Organization: Measuring Alignment (n.d.). SSRN eLibrary, `http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1288485` (retrieved)
7. Huang, C.D., Hu, Q.: Achieving IT-Business Strategic Alignment via Enterprise-Wide Implementation of Balanced Scorecards. Information Systems Management 24(2), 173–184 (2007), doi:10.1080/10580530701239314

8. Goldman, J.E., Ahuja, S.: Integration of COBIT, Balanced Scorecard and SSE-CMM as a strategic Information Security Management (ISM) framework. In: Proceedings of the Fourth International Workshop on Business/IT Alignment and Interoperability (BUSITAL 2009) Held in Conjunction with CAiSE 2009 Conference, Amsterdam, The Netherlands (2009)
9. Goldman, J.E., Ahuja, S.: Integration of COBIT, Balanced Scorecard and SSE-CMM as an Organizational & Strategic Information Security Management (ISM) Framework. In: Quigley, M. (ed.) ICT Ethics and Security in the 21st Century: New Developments and Applications, pp. 277–309 (2011), doi:10.4018/978-1-60960-573-5.ch014
10. Kaplan, R.S., Norton, D.P.: Using the Balanced Scorecard as a Strategic Management System. Harvard Business Review (January-February 1996)
11. Haes, S., Grembergen, W.: IT Governance Structures, Processes and Relational Mechanisms: Achieving IT/Business Alignment in a Major Belgian Financial Group. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS 2005) - Track 8, p. 237b (2005)
12. Balanced Scorecard Institute [BSCI]. About - Balanced Scorecard, `http://www.balancedscorecard.org/BSCResources/AbouttheBalancedScorecard/tabid/55/Default.aspx` (retrieved March 1, 2009)
13. Cobbold, I.M., Lawrie, G.J.G.: The development of Balanced Scorecard as a Strategic Management Tool. 2GC Active Management Ltd. (2002), `http://humanresources.co.za/free/Downloads/BSC.pdf` (retrieved October 18, 2008)
14. Kaplan, R.S.: Using the Balanced Scorecard as a Strategic Management System. Harvard Business Review 74, 75–76 (1996)
15. Kaplan, R.S.: The Balanced Scorecard: Measures that Drive Performance. Harvard Business Review 83(7), 172–173 (2005)
16. Martinsons, M.: The balanced scorecard: a foundation for the strategic management of information systems. Decision Support Systems 25(1), 71–74 (1999)
17. IT Governance Institute. Aligning COBIT® 4.1, ITIL® V3 and ISO/IEC 27002 for Business Benefit (2008), `http://www.isaca.org/Template.cfm?Section=COBIT_Mapping1&Template=/ContentManagement/ContentDisplay.cfm&ContentID=45932` (retrieved February 18, 2009)
18. Larsen, H.M., Pedersen, K.M., Viborg Andersen, V.K.: IT Governance – Reviewing 17 IT Governance Tools and Analysing the Case of Novozymes A/S. In: Proceedings of the 39th Hawaii International Conference on System Sciences (2006)
19. Grembergen, W., Haes, S.: COBIT's Management Guidelines Revisited: The KGIs/KPIs Cascade, vol. 6. IT Governance Institute Publication (2005), `http://www.itgi.org/Template.cfm?Section=Home&CONTENTID=24398&TEMPLATE=/ContentManagement/ContentDisplay.cfm#f3` (retrieved October 6, 2008)
20. Ahn, H.: Applying the Balanced Scorecard Concept: An Experience Report. Long Range Planning 34(4), 441–461 (2001)
21. IT Governance Institute. COBIT Mapping: Mapping SEI's CMM for Software with COBIT 4.0. (2007a), `http://www.isaca.org/Template.cfm?Section=COBIT_Mapping1&Template=/ContentManagement/ContentDisplay.cfm&ContentID=27170` (retrieved December 7, 2008)

22. Gutierrez, A., Serrano, A.: Assessing Strategic, Tactical, and Operational Alignment Factors for SMEs: Alignment Across the Organisation's Value Chain. International Journal of Value Chain Management 2(1), 33–56 (2008)
23. Tan, F.B., Gallupe, B.: Aligning Business and Information Systems Thinking: A cognitive approach. IEEE Transactions on Engineering Management 53(2), 223–237 (2006)
24. Hale, A.J., Cragg, P.B.: Measuring Strategic Alignment in Small Firms. In: Proceedings of the Information Systems Conference of New Zealand, Palmerston North, New Zealand, October 30-31, pp. 127–135 (1996)

Understanding Maturity of Collaborative Network Organizations by Using B-ITa Processes

Faiza Allah Bukhsh[1], Maya Daneva[2], and Hans Weigand[1]

[1] Department of Information Management, Tilburg University, P.O. Box 90153, Tilburg, The Netherlands
{f.a.bukhsh,h.weigand}@uvt.nl
[2] Department of Computer Science, University of Twente, P.O. Box 21, 7500 AE Enschede, The Netherlands
m.daneva@utwente.nl

Abstract. Organization's concerns as controlling costs, improving quality, increasing effectiveness, and managing risk increasingly impose strong requirements on Business-IT alignment (B-ITa). Several Maturity Models came into being for considering improvement actions in B-ITa. The IT-Enabled Collaborative Network organizations (ICoNOs) maturity model specifically addresses the processes needed for achieving B-ITa in networked organizations. We have evaluated ICoNOs maturity model on a case study in an Asian organization. On the basis of this study, we propose to introduce a new domain named as "Cost Management".

Keywords: Business-IT Alignment, Inter-organizational collaboration, Maturity Models, Case Study.

1 Introduction

In the recent years the development of Information Technology (IT) resulted in inventing a variety of collaboration and coordination models based entirely on inter-organizational systems. For communications and collaboration to add value to the business, businesses need to align their business and IT sides. This need is also very well recognized in inter-organizational settings, where multiple businesses or government organizations team up with each other to deliver together a product or a service to clients. In this paper, we adopt the following definition of a network organization: "A network organization is a situation that come up when independent people and groups, linked across boundaries, work together for achieving a common goal" [14,19]. The relationship of network organizations when studied in perspective of business-IT alignment (B-ITa) gives rise to the question of how to measure the maturity of the relationship. To respond to this, researchers have come up with a variety of maturity models (MM)[1,7,15,18,24]. In this research, we focus on one most recent MM, namely ICoNOs [21] developed by Santana Tapia [20]. This model has been evaluated by its author in terms of its suitability on European and American CNOs. However, the previously published evaluation studies are preliminary only and Santana Tapia [20] acknowledged that more evaluation research is needed to confirm the suitability and the

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 580–591, 2012.

usefulness of the ICoNOs model for both government and non-government NOs. In this paper we will take into account the challenge to evaluate the suitability of ICoNOs in a variety of settings, to derive lessons learnt from evaluation study, and to use the learning to enhance the ICoNOs model.

Our paper builds upon the knowledge gained in the early evaluation studies, and sets out to apply the model to a case study in Asia and reflect on the experiences. We carry out an evaluation study in which we attempt to answer the following Research Question: *What is the suitability of the ICoNOs MM to contexts of government Network organizations/ Collaborative Network Organizations (NOs/CNOs) in developing countries for assessing their B-ITa.*

In order to justify the selection of ICoNOs MM in section 2 we have presented a literature review. For evaluating the ICoNOs MM we have selected the qualitative research approach [28]. To do the evaluation, we have selected an Interview based approach, in which we are following a research cycle [31] [32] presented in section 3. Section 4 closes with conclusion and some future research directions.

2 Background

MMs describe evolution of an entity (Organization/ NOs/ CNOs etc). These models are designed for the purpose of assessment and prediction of some attributes for an entity. Literature review done by Santana Tapia's is based on the models especially designed for the B-ITa [20]. In our literature review we explicitly included e-government and e-business models in which B-ITa is one of the key components (but not the only one).

Figure 1 show the literature review we have done on the MMs developed from 2007 till 2009 and e-government models developed from 2001 till 2008. In Santana Tapia's literature review whose scope ranged from 2001 till 2006, the author found that his surveyed MMs mostly relate to B-ITa in a single enterprise and few with B-ITa in CNOs. In case of CNOs the main focus is on processes

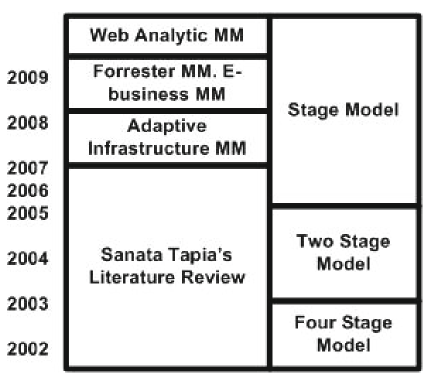


Fig. 1. Results at a Glance

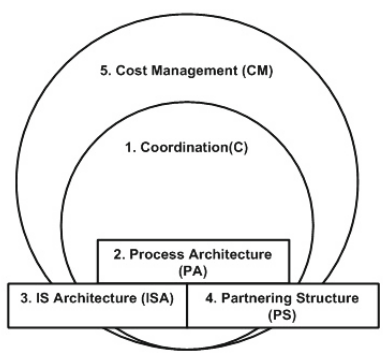


Fig. 2. Relationship of Cost Management with Other Domains

and integration levels (such as SCM MM [15] , E2A MM [24]), relationship (such as IT outsourcing MM [1]), IT sharing and information (such as in extended CMM [17]). We have found that Adaptive infrastructure MM [1] developed especially for the data center management. The well known Forrester MM [18] model address SAAS context while the last model included in the survey from MM's perspective is web analytic MM [7] which describes the maturity of an organization through web analysis. Models reviewed in e-government domain are the stage models which studied e-government growth with respect to multiple stages. With the help of MM, entities become aware of their current state in different areas. Maturity models are usually level based. Domains and levels characterize each MM. Domains define the specific area or direction and levels define the improvement path.

2.1 The ICoNOs Maturity Model

In the following, we will summaries ICoNOs MM domains and levels.

ICoNO's Levels [23] ICoNOs MM consist of five levels, which are used to describe the improvement path for CNOs to achieve B-ITa. These levels are,

- *Level 1:* Incomplete. At this level processes for a B-ITa domain are not performed or partially performed.
- *Level 2:* Isolated. Processes for particular B-ITa domain are planned and executed at initial levels. But these processes are isolated not managed form CNOs perspective.
- *Level 3:* Standardized. Processes are heading for standardization of a particular B-ITa domain from a CNOs perspective.
- *Level 4:* Quantitative Managed. Processes about a particular B-ITa domain use statistical and other quantitative techniques for quality measurement.
- *Level 5:* Optimized. Processes are improved for B-ITa based on common causes and interests.

ICoNO's Domains [23] All the levels defined above are used on certain domain. A domain is a group of processes used to improve a particular area in CNOs. ICoNOs MM consist of four domains which are as follows:

- *Partnering Structure:* Defines where the work will be done and who will be involved. For CNOs it describes organization structure, work division, roles and responsibilities.
- *IS Architecture:* It is about information management function their relationship to each other and the environment. It also describes the principles, guiding the design, and evolution of CNOs.
- *Process Architecture:* Describe about all the individual and collaborative processes needed to reach a shared goal of CNOs.
- *Coordination:* It is about managing the interaction and work among the CNOs while considering the dependencies among the processes.

2.2 Comparing ICoNOs Maturity Model with the Previously Published Work

We must note that the six models presented in Section 2, are all geared to the networked settings. When comparing ICoNOs and any of the six models, we observe that mappings can be found between the ICoNOs design elements and the design elements of the models in our review. For example, the Four Stage Model [12], the HP AIMM [1] and the Forrester model [18] suggest a strong emphasis on processes and integration levels, on relationships, on information sharing, as ICoNOs does. We are however able to point out to areas that fall beyond the scope of ICoNOs. These are: culture, cost management, quality assurance. We trace the inclusion of these areas back to the objectives with which the six models were created. We also think that those areas that are beyond the ICoNOs MM's scope, can be considered as candidates for inclusion in ICoNOs MM, if one considers investigating the possible ways to extend the ICoNOs MM. We make a note of this because it is well possible that replicated evaluations of ICoNOs MM suggest needs for extending it. If this happens, then a researcher would not only use his/her observations coming out of his/her evaluation of ICoNOs MM, but also the observations that are provided through this literature review. Indeed, in the rest of this paper, we will see how we identified the need for one specific enhancement to the original ICoNOs MM and how we responded to this need.

3 Case Study

We have used an interview-based approach for evaluation of ICoNOs MM on an Asian organization (OG).Where OG has one of the largest citizen database. We started by conducting the interviews and concluded that as OG is originally developed by its strategic partners, so the administration don't know the details of technical terminologies evolved behind it. The overall case study plan was inspired by Yin [28] and other researchers [11]. These steps involves,

1. Decide on the MM and Interviewees: We have selected ICoNOs MM as the model to evaluate and OG as an organization for selecting the people to interview.
2. Present the ICoNOs MM to Interviewees: We gave a presentation regarding the ICoNOs MM and describe about the five maturity levels and four domains of ICoNOs. Those are Partnering structure, IS architecture , Process architecture and Coordination.
3. Conduct the Interviews. For the better understanding and convenience of the professionals, most of the interviews were conducted in Urdu (Pakistan's National Language) and then translated into English. In the following section we will discuss the results concluded, We carried out semi-structured interviews based on the interviewing practices recommended in [27]. For the interviews, we used the questionnaire in Santana Tapia's thesis (see Appendix D in [20]). We got an opportunity to meet five professionals. Four

[1] www.hp.com/go/aimm accessed on November 14, 2010

of them were from OG and one was from its local client. In OG our selected professionals were working on the national database of citizens and the one form its strategic client works on the passport authority. In this paper, instead of writing a detailed question-answer-session of each professional, we are dividing the interviews into three parts based on our three broad questions. Below, each question, the views of each interviewee are presented.

3.1 The Data Collection

Below we present samples from our interview data that we collected through our participants. This is to illustrate to the reader the kind of questions being asked, and the type of information being collected. In Section 3.2., we used this information to summarize our findings and draw some conclusions.

Question 1: How OG judges the strengths and weaknesses of the relationship with its partner organizations (namely, its clients)?
1st Interviewee: As OG is the Pakistan's largest database for the citizen, from the start of partnership all the clients change themselves according to OG. If there is some platform incompatibility then it is obligatory for client to adjust with OG. *2nd Interviewee:* OG is the sole provider of citizen's data to other authorities so it has no competitor. Therefore clients have to make the effort to adjust with OG.
3rd Interviewee: OG considers its strong and weak clients by considering the level of coordination that they have among them.
4th Interviewee: Capability of data exchange is the basis for the strong and weak relationship with its clients. For data exchange basic infrastructure needs to be the same. For example, in case of OG and it clients.
5th Interviewee: The requirement document provided by OG to its client is in fact the base of the judgment, for seeing the existing and future possibility of alignment between them.
Question 2: What are the aspects needed, for OG to work successfully with its clients?
1st Interviewee: The coordination on the 'architectural information' is required data.
2nd Interviewee: Cost considerations and quality control are the building stone of the successful relationship. As there exist no competitors for OG they can demand whatever they want.
3rd Interviewee: Coordination and communication among the clients and the OG is one of the major aspects to consider. Lack of communication produces a different outcome then expected or required.
4th Interviewee: Process architecture and business rules and data flows are the most important aspect for OG's success.
5th Interviewee: There exist many aspects but the most important among them are cost management and coordination.
Question 3: Do these recommendations (ICoNOs Maturity Model Domains) suits to OG?

We make the note that the names of the ICoNOs MM domains were not provided in the orientation presentation explicitly, because we wanted the interviewees to give us the necessary domain names from OG themselves. We wanted to see whether the domain names and the earnings that would come up in the minds of the professionals, when thinking about B-ITa, would converge with or diverge from the original names and meanings used in the ICoNOs MM.

1st Interviewee: Maturity consideration with respect to coordination suits to OG but IS architecture and Process architecture are not very important. So we can say that among ICoNOs MM domain coordination and partnership structure (partnering structure in ICoNOs terminology) are basic recommendations.
2nd Interviewee: To measure the maturity of the OG with its clients recommendations such as coordination, partnership structure, IS and business process architecture are important but there should also be some cost consideration.
3rd Interviewee: So far there was no way by which OG can consider maturity aspect and these recommendations provide very practical and convenient framework.
4th Interviewee: Yes, they suit to OG.
5th Interviewee: Not only these recommendations suit to OG but they are also helpful for the clients. Because in this way the client can also find out at which level of maturity he is with OG or with any other organization.

In the following section we will conclude these answers.

3.2 The Results

In our case study research, we came across with following important findings,

- Coordination is fundamental among all the dimensions; we found that any other dimension like Partnering Structure, IS architecture or Process architecture are related to the Coordination. We can say that all the other three domains lie under the cover of Coordination.
- IS architecture and process architecture are being used but at OG, both architectures didn't have any distinct boundaries, in-fact they have overlapping boundaries.

There exist different terms but have same meaning. For example Partnering Structure is equivalent to Partnership Structure, Partnership, CNO is Network Organizations, Partnering Organizations. Based on our learning we confirm the presence of same four key domains (Partnering Structure, IS architecture, Process architecture and Coordination)proposed by Santana Tapia [22]: CNOs involved in B-ITa projects consider these domains for understanding B-ITa maturity. During the interviews three out of four interviews mention additional domains , in particular "Cost Management". But the most important are the four discussed by the ICoNOs MM. During our case study analysis we have considered validity threats states as 'single case study results cannot be generalized to update the ICoNOs MM model'. To confront it, our argument is that not only our case study results favor the introduction of cost domain but the results obtained by Santana Tapia [20] already indicate the importance of cost domain.

3.3 Evaluation of Validity Threats in Interview-based Case Study

We have validated the ICoNOs MM on one of the government organization in Pakistan and found that the ICoNOs MM is equally suitable to Asian government organizations. It is possible to challenge our conclusions by asking a number of questions , which are as follows:

- Was the organization very much developed government organization and, thus, a typical for Asian countries?
- Will ICoNOs MM be equally applicable to other Asian CNOs?
- If we apply ICoNOs MM on other government organizations, will the results be same?

We address these questions in more detail. First, OG is a large provider of government solutions serving many Asian countries and its solution delivery processes are developed and elaborated in a way similar to a solution provider organization in the Western countries (e.g. US or the European countries). This might pose a threat to the generalizability of our results in the sense that they might be specific to large Asian CNOs where one much developed partner (in terms of organizational design) instills its discipline, processes and cooperation rules over its collaborators (in this case, its clients). However, we think that the impact of this threat is minimal as OG seems to be a typical organization in the Asian market of government system delivery. We did a search for evidence on the characteristics (in terms of collaboration) of Asian e-government projects as available in web sites of market research firms in Asia (e.g. Gartner Asia, eGovAsia - www.egovasia.com), the United Nations [26] and the Economist magazine. Specifically, we looked into what kinds of partners are included in Asian vendor and partnership ecosystems. We found that in a number of Asian countries, e-government is built-up with the participation of large solution providers collaborating with country-specific government authorities. We also acknowledge that Asian countries have similar e-government demands and, as it is the case of OG, it is well possible for a solution provider to have as clients the governments of several countries.

Second, we can say that we have selected OG for convenience (in a way randomly) and without any prior analysis. The author of this paper had a previous work relationship with OG and was employed in an internship in a Pakistani government organization. The author was, thus, familiar with the government processes in the country and in the business areas for which OG delivers IT solutions. We acknowledge that other researchers might choose other country's organizations for the reasons similar to ours.

Third, we make the note that the interviews were carried out and transcribed in Urdu language, and also analyzed by one researcher only (namely, the leading author of this paper). Two senior researchers participated in planning the case study, but because they were not familiar with Urdu, there was no way for them to get actively involved in the data analysis. This poses a threat to the accuracy of the data collected and the analysis as the interpretation of the author could have been different from what the interviewees actually wanted to say. This threat was mitigated by a checking step in which others asked questions and attempted to triangulate the data with the information from official documents partly available in English from the web sites of OG and its clients. This triangulation step strengthened the feeling that the level of bias of the first author is reduced.

3.4 Implication of Results from OG: Introduction of the Cost Management Domain

The recommendation obtained by the interviewed professionals from OG suggests that cost is an important factor to consider. Here we compare this finding with what is known in literature about the role of Cost Management in B/ITa, and we motivate why we decided to include it as a new domain into the ICoNOs model.

Indeed, when searching the literature on the role of cost consideration in B-ITa, one can deduce that many theories and models [2,4,5,6,9,10,25],are cost based. The common argument all of these theories and models have is that due to B-ITa cost is being shared among the businesses. Furthermore, the literature survey done by Santana Tapia [20] provides the basis for thinking that cost management is an important dimension of B/ITa. The above findings and our own results motivated us to extend ICoNOs by including the Cost management domain. Our goal of introducing this domain is to improve the completeness and performance of ICoNOs MM. For the purpose of introducing this domain, we set forth a set of questions:

What will be the name of the new domain?
From interviews we have concluded that B-ITa is affected by cost. Among many terms used by OG (Financial Constraint, Cost effects, Cost management, Financial regulations) we have selected “Cost Management” as the name of this new domain [13]. We think that this choice is justified because we found it was intuitive not only to our case study participants but also it converged with the terminology used in the B-ITa literature.

Will the name of the ICoNOs MM be changed after introduction of new domain?
Yes it will change; we will name it as Enhanced ICoNOs MM.

How to define this domain?
The Cost Management domain can be defined as “the discipline of ensuring IT is obtained in the most effective price - which does not necessarily mean cheapest - so that the CNOs can understand the cost of its services and/or products (based on ITIL [10]).

Why is it necessary to introduce it? Cost constraints always affect the progress of an organization. Despite the fact that all the other domains help us to understand the B-ITa, if financial constraints/standards of the partnering organization are not comparable (w.r.t organization's size, type, financial state. etc) then the ICoNOs MM domain may not be enough to make the right decisions.

Will the ICoNOs MM domains be affected by the introduction of a new domain? We think that they will be affected partially. Coordination is the heart of the B-ITa, without it two CNOs can never be aligned. If two CNOs have different cost considerations then how the coordination is possible at equal level? For example, the domain of partnering structure incorporates all aspects related to "IT governance" too. There are always some costs of governance.

Information System architecture and process architecture are also two ICoNOs MM domains and these are partially dependent on cost. These two domains are in fact dependent on information and process flows, information and process flow are affected by cost. From the case study results we found the relationship between the existing domains and new domain. Figure 2 shows relationship of cost domain with ICoNOs MM domains.

We make a note that, in the Figure 2, the relationships among the four ICoNOs MM's domain are borrowed from Santana Tapia's work [23]. We have introduced only the relationship of "Cost Management" domain with ICoNOs MM's domains. Our literature review lets us think that the effect of cost on domains would vary with respect to organization type. According to de Koning and van der Marck [4] the decision of making an investment in an organization is dependent on the organization type/strength/status/maturity. For NOs/CNOs cost is important but how much the cost is important for NOs/CNOs would dependent on this organization's business rules. For CNOs where profit and lost is being shared among the partnering organization [23], cost considerations become important automatically.

3.5 Enhanced ICoNOs MM: Cost Management Domain

Santana Tapia discusses the possibility of the introduction of a new domain in Section 10.3 of his dissertation [20]. ICoNOs MM when evaluated on OG leads us to conclude that "Cost Management" is an important domain.

The introduction of new domain transforms the ICoNOs MM into Enhanced ICoNOs MM. Each domain has its B-ITa processes derived from the theories and models [2,3,5,4,6,9,8,16]. From extensive literature review Santana Tapia selected a set of B-ITa processes that make up the four domains (PS, IS, PA, CO) as shown in Figure 3. For the new domain we used the same literature to identify processes that belong to the Cost Management domain.

The underlined and bold process names show the Cost management domain processes in the modeling map. The following six B-ITa processes are introduced for "Cost Management" domain.

First column of the Table 1 shows the levels of B-ITa processes. It can be argued why we position one process at level 4 and other at level 2. However the decision for such a positioning is derived by the definition of each process.

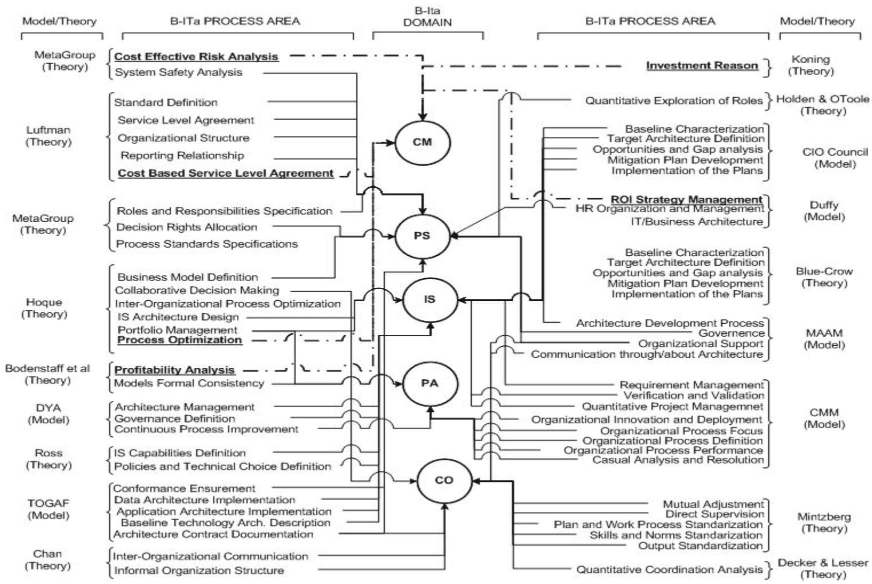


Fig. 3. B-ITa Processes

Table 1. Cost Management Processes

Level	Process	Description
2	Cost-based service level agreements [16]	Describes cost-based cross-organizational ties for decision-making and authority regulation, which in result regulate the CNOs work.
3	ROI strategy and management formulation [5]	Describes the financial management of the organization. In this process, IT costs and benefits are part of general costs and benefits.
4	Process optimization [9]	Describes the principles that support the alignment between business and IT by optimizing the cost related business process.
4	Profitability Analysis [2]	Describes about how a CNO use the profit model in combination with process model and event log to track a snapshot to analyze the profitability of the CNO.
5	Investment reason analysis[4]	This process provides the knowledge to other parties that are external to the CNO. For potential investors, it also gives a reason to invest in the CNO.
5	Cost effective risk analysis [6]	Describes the sources of cost-based problems in business also take the actions to prevent it in future.

4 Conclusion

Maturity Models seems to be a useful vehicle for understanding the alignment of the organization. To the best of our knowledge there exist no MM, which specifically discusses B-ITa in CNOs. The question raised in the Section 1 can be answered by saying that we found that the ICoNOs MM suits the context of Asian CNOs. We also found some similarities between Asian and Western CNOs context regarding the exposure of CNOs partners to disciplined processes. We proposed a new domain as an extension of the ICoNOs MM. It includes six B-ITa processes. Our decision to include these six processes is based on our literature analysis. However, we acknowledge that new scientific studies are being published each month on B-ITa and they may provide good argumentation for including new processes to our set of six. We therefore do not consider our Enhanced ICoNOs MM to be complete and we expect it be extended with more B-ITa processes whose evaluation may possibly add more precision into the final B-ITa maturity assessment itself. Follow-up qualitative case studies are also an important line of future research, so that the findings from literature are triangulated with insights from practitioners. This will build strength in the claim that Cost Management is an important domain to consider when assessing B-ITa maturity. We acknowledge that the maturity assessment is a labor-intensive and time-consuming process as it depends on the involvement of experts. We think that researchers can balance light-weight approaches to maturity data gathering and analysis with more 'interview based' approaches (e.g. interviews). What represents the right balance and how to achieve it remains an open question for the future.

References

1. Adelakun, O., Wabash, S.: IT outsourcing maturity model. In: Proceedings of the 13th European Conference on Information Systems, The European IS Profession in the Global Networking Environment, ECIS, vol. 4 (2004)
2. Bodenstaff, L., Wombacher, A., Reichert, M.: On formal consistency between value and coordination models. Technical Report Centre for Telematics and Information Technology University of Twente (2007)
3. CMMI Team: CMMI for development, version 1.2 (2006)
4. de Koning, D., van der Marck, P.: IT zonder hoofdpijn: een leidraad voor het verbeteren van de bedrijfsprestaties. Prentice Hall (2002)
5. Duffy, J.: Maturity models: Blueprints for e-volution. Strategy & Leadership 29(6), 19–26 (2001)
6. Gunderson, S.: A review of organizational factors and maturity measures for system safety analysis. Systems Engineering 8(3), 234–244 (2005)
7. Hamel, S.: The web analytic maturity model (2009)
8. Henderson, J., Venkatraman, N.: Strategic alignment: Leveraging information technology for transforming organizations. IBM Systems Journal 32(1), 4–16 (1993)
9. Hoque, F.: The Alignment Effect: How to get real business value out of technology. FT Press (2002)
10. Janssen, P.: IT-servicemanagement volgens ITIL. Pearson Education (2005)
11. Johnson, B., Christensen, L.: Educational research: Quantitative, qualitative, and mixed approaches. Sage Publications, Inc. (2007)
12. Layne, K., Lee, J.: Developing fully functional e-government: A four stage model. Government Information Quarterly 18(2), 122–136 (2001)

13. Lee, J.: The scientific endeavor (2000)
14. Lipnack, J., Stamps, J.: The teamnet factor: Bringing the power of boundary crossing into the heart of your business. Wiley (1993)
15. Lockamy III, A., McCormack, K.: The development of a supply chain management process maturity model using the concepts of business process orientation. Supply Chain Management: An International Journal 9(4), 272–278 (2004)
16. Luftman, J.: Assessing IT/business alignment. Information Systems Management 20(4), 9–15 (2003)
17. Ramasubbu, N., Krishnan, M., Kompalli, P.: Leveraging global resources: A process maturity framework for managing distributed development. IEEE Software 22(3), 80–86 (2005)
18. Ried, S., Rymer, J., Iqbal, R.: Forrester's SaaS maturity model: Transforming vendor strategy while managing customer expectations. Forrester Research (2008)
19. Santana Tapia, R.: What is a networked business? Technical Report Centre for Telematics and Information Technology, University of Twente (2006)
20. Santana Tapia, R.: Assessing business-IT alignment in networked organizations. PhD thesis. University of Twente (2009)
21. Santana Tapia, R.: Converging on business-IT alignment best practices: Lessons learned from a dutch cross-governmental partnership. In: ICE 2009: Collaborative Innovation: Emerging Technologies, Environments and Communities, Leiden, The Netherlands (2009)
22. Santana Tapia, R.: ICoNOs MM: The IT-enabled collaborative networked organizations maturity model. Leveraging Knowledge for Innovation in Collaborative Networks, 591–599 (2009)
23. Santana Tapia, R., Daneva, M., van Eck, P., Wieringa, R.: Towards a business-IT aligned maturity model for collaborative networked organizations. In: 12th Enterprise Distributed Object Computing Conference Workshops, pp. 276–287. IEEE (2008)
24. Schekkerman, J.: Extended enterprise architecture maturity model support guide, version 2.0. Institute for Enterprise Architecture Developments (2006)
25. Thomas, R., Tiemann, M., Hite, R., McVay, W., Brundage, G., Knuth, D., Paradis, P., Cullen, T.: Architecture alignment and assessment guide, pp. 1–39. CIO Council (2000)
26. UNDESA. United nations global e-government readiness report 2004: Toward access for opportunity. Technical report (2004)
27. United State Office of Personal Management. Structured interviews: A practical guide
28. Yin, R.: Case study research: Design and methods, vol. 5. Sage publications, Inc. (2009)

Weighted Alignment Measures of Enterprise Architecture Viewpoints

Jacques Simonin[1], Selmin Nurcan[2], and Julie Gourmelen[3]

[1] Institut Mines-Télécom / Télécom Bretagne, Lab-STICC UMR CNRS 6285, UEB, Technopôle Brest-Iroise, 29238 Brest, France
jacques.simonin@telecom-bretagne.eu

[2] Sorbonne Business School, University Paris 1 Panthéon Sorbonne, 90 rue de Tolbiac, 75013 Paris, France
selmin.nurcan@univ-paris1.fr

[3] INSERM, U1018, Université Versailles Saint Quentin, Hôpital Paul Brousse, 16 avenue Vaillant-Couturier, 94800, Villejuif, France
julie.gourmelen@inserm.fr

Abstract. Enterprise Architecture (EA) allows describing how an organization can achieve its objectives and/or develop innovative strategies, through the creation of a set of engineered models that can be understood by the people associated with the organization. To this end, the set of EA models should be developed as any product in any engineering domain. In general, those models are the artifacts on which we reason to understand the alignment between several viewpoints of the organization. Reasoning on those artefacts in an objective and systematic way requires a shared way to represent them and a set of weighted measures based on this way-of-representing. This paper proposes a framework to analyze and to represent multiple viewpoints of an organization and develops a set of measures to qualify the alignment between those models. These measures are experimented from the alignment of Information System functions with respect to business processes for medical research.

Keywords: Enterprise Architecture, Viewpoint, Model, Systemic, Alignment, Measure, Weight.

1 Introduction

1.1 Context and Motivation

Enterprise Architecture (EA) [1] frameworks enable to represent and to analyze multiple viewpoints of an organization. They define an organization system based on those viewpoints. Modeling capabilities are needed to perform this engineering activity and each viewpoint owns its specific set of concepts to this end. In this context, model alignment [2] remains a prerequisite for the organization viewpoints stakeholders who design the organization system using several viewpoint models [3]. A viewpoint describes a partitioning of concepts enabling a modeling of a given system of the organization in relation to this viewpoint [4]. Moreover, thinking in terms of viewpoints allows dividing the system into specific areas of expertise [5].

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 592–599, 2012.

The EA viewpoints alignment definition does not depend on the EA framework choice. We consider in this paper that the alignment between EA viewpoints may be approached from the alignment of models representing each of these viewpoints.

Nevertheless, many problems may prejudice the alignment estimation. In fact, viewpoint models in organization are of varying quality or simply inexistent. Our hypothesis in this work is that *those models exist and they conform to viewpoints meta-modeling*. The alignment is thus defined from this meta-model and applied to viewpoint models.

This paper proposes two new model alignment measures meaning EA viewpoint model concept estimation and viewpoint model global estimation. This estimation is based on a set of quantified measures to score the alignment between the viewpoint models in terms of ratios (values inside the interval [0,1]).

We propose in Section 2 an alignment definition of viewpoint models. In Section 3, the viewpoint models alignment is then evaluated from weighted alignment measures. An epidemiological IS of the French National Institute of Health and Medical Research is the subject of the case study described in Section 4. We conclude in Section 5.

1.2 Related Work

Most works on the alignment measurement of EA viewpoint models highlight organization need of business – IS consistency enabling organization results improvement [6]. To satisfy this need, one solution is to compare its 'enterprise viewpoint' (including strategy) and its IS or IT viewpoints. From this alignment, heuristics have been defined, for instance, to provide warnings in case of misalignment [7]. Measurement methods concerning EA allow evaluation of this architecture in business terms (cost, benefit, risk impact analysis).

Other works about alignment measurement are based on strategic elements compared to operational elements [8]. Alignment measure is then based on quality criteria and measure ontology [9].

Significant works specific to IS model alignment with respect to organization strategy choose fitness measures [10]. These measures are based on alignment ratio. The quantity of business concepts well aligned with IS concepts is thus divided by the quantity of business concepts. Similar measures are proposed in [11] to estimate the alignment of an IS model with respect to a business model. This estimation concerns the 'computational viewpoint' with respect to the 'enterprise viewpoint', including the organization entities and their relationships.

Our previous work [12] about viewpoint models alignment proposes axiomatized measures based on such alignment ratios. Those ratios are between the quantity of well aligned elements from 'information viewpoint' with respect to 'enterprise viewpoint' and the quantity of 'information viewpoint' elements.

2 EA Viewpoint Alignment Definition

We propose to complete the EA viewpoint modeling process with an alignment measure, whatever the viewpoint. EA viewpoint modeling process consists in the design of one or more viewpoints that build upon the EA.

2.1 Alignment Definition Based on Systemic Approach

In this paper, an alignment between two viewpoints is equivalent to the alignment between some models representing each aligned viewpoint [13]. Viewpoint models alignment needs concept mapping. We propose a systemic approach to define the models concepts. In the rest of the paper, a concept is so a unit or a relationship between units enabling their interaction.

The alignment is first defined for a MM_V_W work viewpoint meta-model with respect to a MM_V_R reference viewpoint meta-model by:

$$F_A : MM\_V_W \rightarrow MM\_V_R$$

where one MM_V_W concept (a MM_V_W unit or a MM_V_W unit relationship) is associated to a MM_V_R concept in the viewpoints meta-model.

2.2 Viewpoint Models Units' Alignment Application

We propose an alignment function, i.e. f_A, which conforms to F_A. The alignment objective is a mapping of a meta-model concept instance of a work viewpoint model with respect to meta-model concept instances of a reference viewpoint model. The units' alignment of the M_V_W work viewpoint model with respect to the M_V_R reference viewpoint model is defined by:

$$\begin{aligned} f_A : M\_V_W &\rightarrow \mathrm{P}(M\_V_R) \\ u_W &\mapsto \{u_{Ri}; u_{Ri} \text{ aligned with } u_W\}_{i\in[1..n]} \end{aligned} \text{, where}$$

- $\mathrm{P}(M\_V_R)$ is the power set of M_V_R,
- u_W is a M_V_W unit, u_{Ri} a M_V_R unit,
- M_V_W conforms to MM_V_W meta-model and M_V_R conforms to MM_V_R meta-model,
- n (that could be 0) is the quantity of the M_V_R units aligned with u_W.

2.3 Viewpoint Models Relationships' Alignment Application

Alignment between unit relationships only takes into account work viewpoint units, which are aligned with at least one or more units of the reference viewpoint. We extend so the 'f_A' definition to the relationships of the viewpoint models:

$$\begin{aligned} f_A : M\_V_W &\rightarrow P(M\_V_R) \\ ur_W &\mapsto \left\{\begin{matrix} ur_{Ri}; (ur_{Ri} = (u_{Ria}, u_{Rib}) \text{ aligned with } ur_W = (u_{Wa}, u_{Wb})) \wedge \\ (u_{Ria} \in f_A(u_{Wa})) \wedge (u_{Rib} \in f_A(u_{Wb})) \end{matrix}\right\}_{i\in[1..m]} \end{aligned} \text{, where}$$

- $P(M\_V_R)$ is the power set of M_V_R reference model,
- ur_W is a M_V_W work model unit relationship between u_{Wa} and u_{Wb} units,
- ur_{Ri} is a M_V_R unit relationship and
- m is the quantity of the M_V_R unit relationships aligned with ur_W.

3 EA Viewpoint Alignment Measure

Two measures of the alignment between viewpoint models are proposed. The first one enables to estimate the alignment of one concept of the work viewpoint model with respect to the reference viewpoint model. The second one provides an estimation of the alignment of the work viewpoint model with respect to the reference viewpoint model.

3.1 Viewpoint Model Concept Alignment Measure

A concept alignment measure is first proposed thanks to an axiomatization [14]. An axiom is an expected and understandable property of a concept alignment measurement. The following concept alignment (**CA**) axioms define the intuitive behavior of the alignment of one concept sc_W of a work viewpoint model with respect to a reference viewpoint model. The first principle of the alignment behavior is that the alignment depends on the quantity of elements of the $f_A\ (sc_W)$ set. This quantity is represented by the function *Card* in this paper.

CA1 – One-to-one alignment. An alignment of one sc_W concept (unit or relationship between units) of a M_V_W viewpoint model with respect to a M_V_R viewpoint model is perfect if the concept is aligned with only one concept of the M_V_R viewpoint model.

A perfect alignment enables to define without any ambiguity a concept of the work viewpoint model with respect to a concept of the reference viewpoint model.

CA2 – One-to-many alignment. An alignment of one sc_W concept (unit or relationship between units) of a M_V_W viewpoint model with respect to a M_V_R viewpoint model is better if $Card(f_A(sc_W))$ decreases without reaching zero and is worse if $Card(f_A(sc_W))$ increases.

CA3 – One-to-zero alignment. An alignment of one sc_W concept (unit or relationship between units, but not both) of a M_V_W viewpoint model with respect to a M_V_R viewpoint model is null if the concept is not aligned with any concept of M_V_R.

The following concept weighted alignment (**CWA**) axiom defines the intuitive behavior of a weighted alignment.

CWA – Concept weighted alignment. A weighted alignment of a concept sc_W (unit or relationship between units) of a M_V_W viewpoint model with respect to a M_V_R viewpoint model decreases if the weight of sc_W decreases and increases if the weight of sc_W increases.

CWA highlights that the alignment of a concept that is not significant for a viewpoint stakeholder of the organization must be investigated later than a prime one.

A μ_{CWA} measure of the weighted alignment of the sc_W concept such as $sc_W \in M\_V_W$ with respect to a M_V_R viewpoint model could be the following:

$$\begin{aligned}&\mu_{CWA}(sc_W / M\_V_R) = \omega_W(sc_W) \times \left(\frac{1}{Card(f_A(sc_W))} \right), \text{if } Card(f_A(sc_W)) \neq 0. \\ &\mu_{CWA}(sc_W / M\_V_R) = 0, \text{if } Card(f_A(sc_W)) = 0\end{aligned} \tag{1}$$

where:

- M_V_R is the set of concepts (unit if sc_W is a unit, relationship between units if sc_W is a relationship between units) of the reference viewpoint model.
- $\omega_W : M\_V_W \to [0,1]$ associates a weight to a concept of the M_V_W model.

3.2 Viewpoint Models Weighted Alignment Measure

The alignment of a work viewpoint model with respect to a reference viewpoint model provides an estimation of the consistency between the two viewpoint models. The following axiom (**GWA**) defines the intuitive behavior of this weighted alignment deduced from the alignment axiomatization.

The principle of the weighted alignment behavior is that the weight of sc_W and the weight of the $f_A(sc_W)$ set constrain the alignment. This constraint means that the global weighted alignment estimation of each concept is balanced by these weights (weighted average).

GWA – Global Weighted Alignment. The weighted alignment between the concepts (units or relationships between units, but not both) of a M_V_W work viewpoint model with respect to a M_V_R reference viewpoint model is estimated by the weighted average of all the non-weighted alignment estimation.

A μ_{GWA} measure of the weighted alignment of M_V_W with respect to M_V_R consistent with (**1**) and with the **GWA** axiom could be:

$$\mu_{GWA}(M\_V_W / M\_V_R) = \frac{\displaystyle\sum_{\left\{\substack{sc_W \in M\_V_W;\\ Card(f_A(sc_W)) \neq 0}\right\}} \left(\omega_W(sc_W) \times \frac{\sum_{sc_R \in f_A(sc_W)} \omega_R(sc_R)}{Card(f_A(sc_W))} \times \frac{1}{Card(f_A(sc_W))} \right)}{\displaystyle\sum_{\{sc_W \in M\_V_W\}} \left(\left(1_{\left\{\substack{sc_W \in M\_V_W;\\ Card(f_A(sc_W)) \neq 0}\right\}}(sc_W) \times \omega_W(sc_W) \times \frac{\sum_{sc_R \in f_A(sc_W)} \omega_R(sc_R)}{Card(f_A(sc_W))} \right) + \left(1_{\left\{\substack{sc_W \in M\_V_W;\\ Card(f_A(sc_W)) = 0}\right\}}(sc_W) \times \omega_W(sc_W) \right) \right)}. \tag{2}$$

where:

- 1_{M_V} is the indicator function defined on the M_V model concepts set;
- M_V is a set of model units nor a set of model relationships between units;
- $\omega_W : M\_V_W \to [0,1]$ associates a weight to a concept of the M_V_W model;
- $\omega_R : M\_V_R \to [0,1]$ associates a weight to a concept of the M_V_R model.

4 Real Case Study of Enterprise and Information Viewpoints Alignment

We choose in this real case study the Reference Model for Open Distributed Processing (RM-ODP) framework [15]. The case study is the alignment of an ‘information viewpoint’ with respect to the ‘enterprise viewpoint’ that emphasizes the consistency of an IS with the supported organization’s business. The IS takes place in an epidemiological research unit of INSERM, the French National Institute of Health and Medical Research [16]. The axioms proposed in the Section 3 are useful in this real study case because the vocabulary of the ‘enterprise viewpoint’ is different from the vocabulary of the ‘information viewpoint’. Indeed, the epidemiology researchers use on the one hand a medical vocabulary in the ‘enterprise viewpoint’ and the statisticians or the data request designers use on the other hand a technical vocabulary in the ‘information viewpoint’. A one-to-one alignment (see **CA1** in the Subsection 3.1) is then useful to validate the association of every term of the technical vocabulary with one term of the medical vocabulary.

4.1 Alignment Definition

The alignment study case is restricted to meta-model units. For the reference ‘enterprise viewpoint’ ($V_R = V_E$), the unit is the business activity, which composes a business process, as unit (24 activities compose 3 business processes dealing with the epidemiological studies).

The concepts chosen for the work ‘information viewpoint’ ($V_W = V_I$) is the information block composed of functions that support a business activity (21 blocks designed in the epidemiological IS).

The alignment of the information blocks with respect to the business activities is justified by a relationship *supports* (specified inside the viewpoints meta-model of the INSERM) meaning that one information block may support one or more business activities.

4.2 Information Block Weighted Alignment Measure

The INSERM stakeholders have to apply first the units’ alignment (see Subsection 2.2) of an information block with respect to the business activities. In this case study, the weight is deduced from a categorization of the information blocks. An information block category defined in [12] is ‘Command’ (for example, *IBAccessRightManagement* specific to one epidemiological system use) or ‘Data’ (for example, *IBCatalogueVariableManagement* transverse to more than one epidemiological system uses). The weight of an information block is 1.0 if ‘Data’ categorized or 0.5 if ‘Command’ categorized.

One alignment illustration is for example the *IBCatalogueVariableManagement* catalogue management information block that supports nine business activities: *Provide data*, *Design the database*, *Map data*, *Filter data*, *Agree the exchange format*, *Give an access to the database*, *Check the database*, *Validate the scientific conformity*, *Validate the technical conformity*. *IBCatalogueVariableManagement* is moreover categorized as a 'Data' information block ($\Rightarrow$ ω_I({*IBCatalogueVariableManagement*}) = 1.0),

$\Rightarrow$ μ_{CWA} ({*IBCatalogueVariableManagement*}/V_E) = 1.0 x (1/9) = 0.11 (see (**1**)).

The ranking has a beneficial effect during the alignment study of the 'information viewpoint' with respect to the 'enterprise viewpoint'. It enables indeed a prioritization of the information blocks for the alignment improvement activity.

4.3 Information Viewpoint Weighted Alignment Measure

For the global weighted alignment measure (see Subsection 3.2), the useful weights are on the one hand a weight associated to the concepts of the V_I work 'information viewpoint' (see Subsection 4.2) and on the other hand a weight associated to the concepts of the V_E reference 'enterprise viewpoint'.

In this case study, the weight associated to the concepts of the 'enterprise viewpoint' underlines that a business activity specific to the INSERM is more important than an activity external to the INSERM for the stakeholders.

The result of the estimation of the weighted alignment measure of the 'information viewpoint' units' alignment with respect to the 'enterprise viewpoint' is (see (**2**)):

$$\mu_{GWA}(M\_V_I / M\_V_E) = 0.47$$

This weighted alignment measures of the 'information viewpoint' alignment with respect to the 'enterprise viewpoint' highlights the non-fulfillment of the business processes description by the INSERM.

5 Conclusion

The systemic approach enables to apply simply alignment between viewpoint models in relation with any EA framework. The alignment application between the unit relationships of the work viewpoint model and the unit relationships of the reference viewpoint model may be moreover automatically deduced from the units' alignment application.

Our real case study highlights nevertheless the complexity of the alignment behavior. The CWA concept weighted alignment axiom and the GWA global weighted alignment axiom introduce some knowledge specific to the viewpoint stakeholders. Relevant weights selected by the viewpoint stakeholders are one possibility enabling the improvement of the alignment estimation accuracy between viewpoint models. Decision aid could be an interesting approach to refine the weight of every viewpoint model element.

References

1. Zachman, J.A.: A Framework for Information Systems Architecture. IBM Systems Journal 26(3), 276–292 (1987)
2. Henderson, J.C., Venkatraman, N.: Strategic Alignment: Leveraging Information Technology for Transforming Organizations. IBM Systems Journal 32(1), 4–16 (1993)

3. Chen, P.P.: Suggested Research Directions for a New Frontier – Active Conceptual Modeling. In: Embley, D.W., Olivé, A., Ram, S. (eds.) ER 2006. LNCS, vol. 4215, pp. 1–4. Springer, Heidelberg (2006)
4. IEEE Standards Association: IEEE Std 1471-2000 - IEEE Recommended Practice for Architectural Descriptions for Software-Intensive Systems-Description (2000), http://standards.ieee.org/reading/ieee/std_public/description/se/1471-2000_desc.html
5. Barkmeyer, E.J., Feeney, A.B., Denno, P., Flater, D.W., Libes, D.E., Steeves, M.P., Wallace, E.K.: Concepts for Automating Systems Integration. National Institute of Standards and Technology (2003)
6. Sabherwal, R., Chan, Y.E.: Alignment between Business and IS Strategies: a Study of Prospectors, Analyzers and Defenders. Information Systems Research 12(1) (2001)
7. Pereira, C.M., Sousa, P.: Enterprise Architecture: Business and IT Alignment. In: ACM Symposium on Applied Computing, Santa Fe, New Mexico, USA (2005)
8. 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 and WES 2007. LNCS, vol. 4495, pp. 203–217. Springer, Heidelberg (2007)
9. Ruiz, F., Genero, M., García, F., Piattini, M., Calero, C.: A Proposal of a Software Measurement Ontology. In: Conference on Computer Science and Operational Research, Buenos Aires, Argentina (2003)
10. 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)
11. Vasconcelos, A., Periera, C.M., Sousa, P., Tribolet, J.: Open Issues on Information System Architecture research domain: the Vision. In: International Conference on Enterprise Information System (2004)
12. Simonin, J., Bertin, E., Le Traon, Y., Jézéquel, J.-M., Crespi, N.: Analysis and Improvement of the Alignment between Business and Information System for Telecom Services. International Journal on Advances in Software 4(1&2) (2011)
13. Lopes, D., Hammoudi, S., Abdelouahab, Z.: Schema matching in the context of model driven engineering: From theory to practice. In: Advances in Systems, Computing Sciences and Software Eng. Springer, Netherlands (2006)
14. Shepperd, M.J., Ince, D.: Derivation and Validation of Software Metrics. Oxford University Press (1993)
15. RM-ODP: ITU-T Recommendation X.901 to X.904 | ISO/IEC 10746-1 to 4 and X.911 | ISO/IEC 15414, http://www.rm-odp.net/
16. Goldberg, M., Quantin, C., Guéguen, A., Zins, M.: Bases de données médicoadministratives et épidémiologie: intérêts et limites. Courrier des Statistiques (124) (2008) (in French)

Enterprise Coherence in the Dutch Ministry of Social Affairs and Employment*

Roel Wagter[1,3], H.A. (Erik) Proper[2,3], and Dirk Witte[4]

[1] Ordina, Nieuwegein, The Netherlands
[2] PRC Henri Tudor, Luxembourg
[3] Radboud University Nijmegen, Nijmegen, The Netherlands
[4] Logica, Amstelveen, The Netherlands
roel.wagter@ordina.nl, erik.proper@tudor.lu, dirk.witte@logica.com

Abstract. This paper is concerned with a real world case study in Business/IT alignment at the strategic level. The case study is situated in the Dutch public sector, involving the Ministry of Social Affairs and Employment (SAE). In this case study, the GEA (Generic Enterprise Architecting) method was used. This paper will therefore take the GEA method as a given. Nevertheless, to better understand and appreciate the case study, we will also briefly review the GEA method and its background. Even more, we will also provide an evaluation on the GEA method, which was/is developed using a design science approach.

Keywords: business-IT alignment, enterprise coherence, enterprise architecture.

1 Introduction

This paper is concerned with a real world case in Business/IT alignment at the strategic level. The case is situated in the Dutch public sector, involving the Ministry of Social Affairs and Employment (SAE). It concerns the introduction of a new system for the creation of a digital document/dossier flow. The introduction of this system was a direct consequence of a government decision to automate these document processes by 2015. It was decided by the Ministry to re-use the system that was already designed, and built, to support similar processes at another Ministry (the Ministry Internal Affairs and Kingdom Relationships). Therefore, the focus of the case is not so much on the creation of a new solution, but rather on the impact on the existing organization when using an existing solution. The specific business issues addressed in the case are: (1) *What are the necessary change initiatives needed for the introduction of this new system?* (2) *What are the best choices in terms of solution direction and approach?*

In the case study, the GEA (Generic Enterprise Architecting) method was used. Given the focus of this paper on the actual case study, we take the GEA method as a given. Nevertheless, to better understand and appreciate the case study, we will also briefly review the GEA method and its background.

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

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 600–607, 2012.

The remainder of this paper is structured as follows. In Section 2, we provide more background to the GEA method. Section 3, then continues by summarizing that part of the GEA method that is most relevant to understanding the case study: the Enterprise Coherence Framework (ECF), which enables enterprise to set up their own *coherence dashboard* in terms of which the enterprise coherence can be governed/improved during enterprise transformations. The SAE specific configuration of this *dashboard* is discussed in Section 4. This is followed in Section 5, by a discussion on how this dashboard was used in a workshop to tackle the business issue at hand. Before concluding, Section 6 then briefly discusses an evaluation by the participants of the case.

2 Background to the GEA Method

The development of the GEA (General Enterprise Architecting) method was initiated in 2006 by the consultancy firm Ordina (www.ordina.nl). The decision by Ordina to initiate the development of the GEA method originated from the observation that large scale enterprise transformations fail more often than not, while, in their experience, existing methods and frameworks for enterprise architecture failed to contribute to the success of enterprise transformation efforts [6, 5]. A survey held at the start of the GEA programme, showed that this experience was not limited to Ordina only, but was shared among a broad range of client organizations participating in the programme[1]. The underlying issues were also considered grave enough for the participating client organizations to indeed co-invest, in terms of time and money, in the GEA programme. In the (still ongoing) development of GEA, the design science method [3] is used as the overarching "rhythm", combined with case study research [7] to evaluate the application of the different iterations of the GEA method.

In its current form, the GEA method comprises three core ingredients [5]. Next to the Enterprise Coherence Assessment (ECA) that allows organizations to assess their ability to govern coherence during enterprise transformation, it contains an Enterprise Coherence Framework (ECF) and a (situational) Enterprise Coherence Governance (ECG) approach. The latter includes the identification of specific deliverables to produced/results, processes needed to produce these deliverables/results, as well as an articulation of the responsibilities and competences of the people involved. The ECF, which will be summarized in the next section, enables enterprise to set up their own *coherence dashboard* in terms of the enterprise coherence can be governed/improved during enterprise transformations. This, enterprise specific, dashboard enables senior management to govern the coherence between key aspects of an enterprise during a transformations.

[1] During different stages of the GEA research programme, the following client organizations were involved: ABN AMRO; ANWB; Achmea; Belastingdienst - Centrum voor ICT; ICTU; ING; Kappa Holding; Ministerie van Binnenlandse Zaken en Koninkrijksrelaties; Ministerie van Defensie; Ministerie van Justitie - Dienst Justitiële Inrichtingen; Ministerie van LNV - Dienst Regelingen; Ministerie van Landbouw, Natuur en Voedselkwaliteit; Nederlandse Spoorwegen; PGGM; Politie Nederland; Prorail; Provincie Flevoland; Rabobank; Rijkswaterstaat; UWV; Wehkamp (see also www.groeiplatformgea.nl).

3 The Enterprise Coherence Framework

The Enterprise Coherence Framework (ECF, see [5]) defines a series of cohesive elements and cohesive relationships, which together define the playing field for an enterprise's coherence. By making the definition of these elements explicit in a specific enterprise, a *coherence dashboard* results in terms of which one can gain insight in the 'state of coherence' while also being able to assess the impact of potential/ongoing transformations. This then enables a deliberate governance of enterprise coherence during/driving transformations.

The ECF is defined in terms of two connected levels of cohesive elements: the level of *purpose* and the level of *design*. At the level of purpose, the cohesive elements that have been identified, correspond to the commonly known concepts from strategy formulation [4, 1]: *Mission*, *Vision*, *Core Values*, *Goals* and *Strategy*. The cohesive elements at the design level are:

Perspective – an angle from which one wishes to govern/steer/influence enterprise transformations. The set of perspectives used in a specific enterprise depend very much on its formal and informal power structures. Both internally, and externally. Typical examples are culture, customer, products/services, business processes, information provision, finance, value chain, corporate governance, etc.

Core concept – a concept, within a perspective, that plays a key role in governing the organization from that perspective. Examples of core concepts within the perspective Finance are, for instance, "Financing" and "Budgeting".

Guiding statement – an internally agreed and published statement, which directs desirable behaviour. They only have to express a desire and/or give direction. Guiding statements may therefore cover policy statements, (normative) principles [2] and objectives.

Core model – a high level view of a perspective, based on, and in line with, the guiding statements of the corresponding perspective.

Relevant relationship – a description of the connection between two guiding statements of different perspectives.

The presence of a well documented enterprise mission, vision, core values, goals and strategy are preconditions to be able to determine the content of the core factors on the design level of the organization.

4 The Coherence Dashboard for the Ministry of SAE

Since this was the first time for the Ministry of SAE to apply/use the GEA method, it was necessary to first develop an organization specific coherence dashboard. To this end, the case at the Ministry of SAE started in August 2010 with an intensive desk research activity, conducted by a small team of architects. This team studied relevant policy documents from the Ministry of SAE, resulting in the first version of the coherence dashboard for the Ministry, in terms of a list of the cohesive elements and their definitions, covering both the purpose and the design level. Starting point for creating this list were the strategic documents of the organization such as the mission statement, vision notes, policy plans, business strategy, business plan, etc.

In a validation workshop, conducted in September 2010, this draft coherence dashboard was then validated with the major stakeholders and approved after some modifications. This validation workshop involved the executives of the Ministry, complemented with a number of (internal) opinion leaders and key stakeholders.

Table 1. Definitions of perspectives for the Ministry of SAE

Perspective	**Definition**
Information provisioning	All processes, activities, people and resources for obtaining, processing and delivery of relevant information for SAE.
Collaboration	Collaboration needed to contribute to a common result on the team, entity or organization levels.
Processes	A coherent set of activities needed to deliver results of SAE.
Governance	The influencing of the SAE organization so that a desired goal is attained.
Employees	All persons who execute tasks or activities within the SAE organization.
Stakeholders	Legal entities or persons for whom the activities of SAE are important.
Culture	Explicit and implicit norms, values and behaviour within the SAE organization.
Services	All services that SAE within legal frameworks, or through agreed appointments with statutory authorities, establishes and delivers to customers.
Finance	The planning, acquisition, management and accountability of funds SAE.
Customers	Customers of a service of SAE
Law & regulations	All legal frameworks that form the basis for the task performance of SAE.
Communication	An active process in which information is exchanged between two or more parties or persons, regardless of how that is achieved.

In Table 1, the perspectives that were selected by the Ministry of SAE are shown, while the core concepts of four of the perspectives are listed in Table 2. This set of perspectives also illustrates the need to align more aspects of an enterprise rather than just business and IT. Several of the perspectives may put *requirements* towards IT support, *information provisioning* followed by *communication* being the dominant ones in this sense. However, the chosen set of perspectives shows that when it comes to *alignment*, the stakeholders do not think in terms of Business/IT alignment, but rather in a more refined web of aspects that need alignment.

Table 2. Core concepts for the Ministry of SAE

Information provision	Processes	Governance	Stakeholders
Digitization	Time and place independent	Policy cores	Labor market
Integrality	Selection policy	Programs	Municipalities
Security	Efficiency	Scaling up	Labor force
Standardization	Actor	Collectivity	Employers Unions
Facilities	Effectiveness	Mission/vision assessment	Employee Unions
Information	Predictability	Employer ship	Funds
Maintenance	Planned	Themes and tasks	Other Ministries
Systems	Procedures	Functioning	Independent administrative bodies
Ownership		Organization	Society
Storage			Social and Economic Council
Architecture			Research agencies
			Social partners
			National Archive

Table 3. Guiding statements relevant to the *processes* perspective

Processes
A dual situation in which paper and digital systems or more systems are used in parallel, should where possible be avoided.
SAE is based on the tenet that the entire work of staff and processflow of documents goes digital.
The concept of flexible working means customization (instead of one size fits all).
Existing paper-based processes of SAE are as much as possible adjusted to the features of the automated document management system.
Integral approach: It is important to think about sustainability already at the "front" of the information chain.
Selection policy must play a fully involved role at the beginning of the "information creation".
The coming years it is expected that firm pressure will be on the business operations and IT to operate cost-efficiently.
Working smarter with fewer people.
We aim to ensure the government can operate decisively, transparently and fast.
We involve at the front of the process the external actors in the issues and developments we are working on.
We must have more attention to the process.
In 2012, our work is supported by a modern work environment and we as professionals SAE are equipped to let this environment operate as optimal as possible for us.
We want better performing processes, more efficient and effective.
We want more predictability in our processes.
It must be clear how processes flow through the organization and who has which responsibilities.

During the desk research phase 219 guiding statements were derived from the aforementioned policy documents. Needless to say that presenting all 219 guiding statements goes beyond the purpose of this paper. Therefore, Table 3 only shows those guiding statements that turned out to be relevant to the *processes* perspective.

5 The Process Followed in the Case Study

With the dashboard in place, the next step was to organize a workshop with the key stakeholders. In this workshop, the business issue at hand (*the introduction of a new system for the digitization of the flow of dossiers*) was positioned in relation to the coherence dashboard of the Ministry of SAE, and analysed in terms of the two questions: (1) *What are the necessary change initiatives needed for the introduction of this new system?* and (2) *What are the best choices in terms of solution direction and approach?*

During the workshop, each of the twelve perspectives of Table 1 was represented by one or two participants who had (delegated) ownership of that perspective. At the start of the workshop, the owner of the business issue gave a thorough introduction of the issue in terms of causes, degree of urgency, degree of interest, implications, risks, etc. See Table 4 (these lists were also handed out to the participants, before the workshop took place). This introduction gave the representatives of the perspectives a deeper insight into the associated aspects of this business issue, enabling them to make a translation of the issue to their own perspective. This enabled the representatives of the different perspectives jointly determine, which change initiatives were required to solve the business issue at hand. The business issue: "*impact of the implementation of a digitization solution*" was then addressed in terms of two tasks: (1) *Determine the necessary change initiatives based on the analysis of the business issue* and (2) *Determine the necessary change initiatives based on the solution space dictated by the guiding statements of the coherence dashboard of the Ministry* (such as for example shown in Table 3).

Table 4. Part of the analysis of the business issue in terms of causes, implications and risks

Causes to adopt a digitization solution			
	1	Government conducts restrictive policy for ICT investments.	
	2	Government wants rapidly resolve many issues in the field of archives, digital information and cultural heritage:	
	3	a	No view on growth, size and cost of archiving.
		b	Issues are already playing for three decades.
		c	Government Decision: digital document management in the core departments by 2015.
	4	Interdepartmental cooperation	
	5	In the field of archiving:	
		a	Many copies and versions.
		b	Many documents are missing.
		c	Rules and compliance are inadequate in the field of digitization.
		d	Digitization is focused on storage and not to reuse.
	6	In the field of processing (dossier flow)	
		a	Not timely delivery (including emergency notes, pieces of Ministers)
		b	Many errors in submission, registration and also in the content.
		c	Ambiguous differentiation of dossiers (Name, Address, City)
		d	The author of a document is difficult to reach (especially with emergency items
		e	Errors far too late in the process discovered.
		f	Lack of adequate information and proper use.
		g	Lack of good management information (where, who, when, how long).
Implications of the digitization solution			
	1	The employee gets a central position.	
	2	Incoming physical mail digitized and only processed digitally.	
	3	Office Documents in digitizing system created and to use by colleagues.	
	4	Other media (e-mail, sound, photographs, video) are stored.	
	5	Never (older) texts lost.	
	6	One organization-wide environment for the dossier flow.	
	7	All documents in dossiers accessible to everyone, unless …	
	8	Managers will be active users by digitally agreeing.	
	9	The entire process is visible to everyone.	
	10	The initials line will be standardized within the own organizational unit.	
	11	There shall be no "co initials" anymore.	
	12	Employees will carry out all work with documents by using the digitization system (except Inspection and Legislation).	
Risks by implementing the digitization solution			
	1	Low acceptance of the user, because too much from the ICT is argued.	
	2	No conscious guidance on quantitative benefits, because the business case does not give this insight.	
	3	Subjective assessment of the results by no clear purpose.	
	4	Errors by improper use.	
	5	Errors due to complex procedures (due to many exception rules).	
	6	Not a good government of the dossier flow by confusion of responsibilities and no control.	
	7	Not learning from mistakes by taking over behavior.	
	8	Not learning from mistakes by failing want to be addressed.	
	9	Final results of the dossier flow are not achieved due to the gap between directors and senior staff.	
	10	Employees do not indicate errors to each other due to lack of management support.	
	11	Suboptimal solution by limited (financial) resources.	
	12	Additional customization because specific management steps do not fit together.	
	13	No broad accessibility and standardization by different solutions for the same functionalities	
	14	Low commitment and support due to poor communication to stakeholders	
	15	Project failure due to lack of management attention.	

Prior to this workshop, all 22 representatives of the perspectives received a copy of: an overview of all the perspectives and core concepts (see Table 2) and their definitions, an overview of the 219 guiding statements including the perspectives they are connected to, a list of guiding statements on each one perspective (see example Table 3), and a summary of the business issue at hand (see Table 4). In addition, two input forms were issued for the two of the tasks that would need to be performed during the workshop. After the introduction of the business issue by the problem owners, the group of 22 persons was split into four subgroups balanced in proportion to the number of guiding statements and the grouping of perspectives with a strong mutual relevant resemblance. The groups were located in different project rooms and asked to give a plenary wrap up by discussing their three major change initiatives after carrying out the three following tasks:

1. *Change initiatives based on the analysis of the business issue*
   The group was asked to use the causes, implications and risks as identified in Table 4, to list the necessary change initiatives in their perspective.
2. *Change initiatives based on the guiding statements*
   The group was asked to, based on the guiding statements, list those change initiatives that could be carried out to solve/mitigate aspects of the business issue.
3. *Prepare for plenary wrap-up*
   After performing both tasks, each group was asked to identify the three major change initiatives, and prepare a presentation of these initiatives as input to the next plenary part of the workshop.

The workshop resulted in 98 change initiatives of which 15 were prioritized as most important ones. In the last plenary part of the workshop these major change initiatives were presented and all the attendees were offered the opportunity to comment on these. The workshop results were presented as an advisory report to the management of SAE, to decide on the proposed solution and approach.

6 Experiencies and Insights for Improving GEA

At the end of the workshops, an evaluation session was organized with the participants of the workshops. This evaluation session resulted in the following shared observations:

1. The participants of the workshops already knew the key principles of this case, but especially the confrontation of these principles with the intended objective of the change program, and the discussions about this were regarded as useful. This provided support, management awareness as well as a more complete picture.
2. An acceleration of the decision-making process and the creation of support at the board level was achieved.
3. A much more holistic approach to the issue compared to the current IT-driven approach. This led to the recognition that much more needed to be changed in the organization than previously assumed.
4. A shorter lead time for obtaining the perspectives and core concepts as a result of the strategy used to first derive guiding statements from policy documents.
5. Saving a lot of processing time regarding the elaboration of the workshop results due to the use of digital forms. This was also experienced as a pleasant way of working by the representatives of the perspectives.
6. The turnaround time of developing the outline of a solution direction, and the choice of the approach to be taken, was reduced to one day using the coherence dashboard.
7. Only a limited number of SAE-employees, for a limited amount of time, (3 hours validation session and 6 hours workshop analyses) were needed in applying GEA.
8. The experience of having 22 representatives of the perspectives meet in a workshop requires timely planning and a convincing modus operandi from the project team, based on a clear problem solving vision and arguments based on added value.
9. The business issue at hand should be positioned at the right management and priority level. This may sound trivial, but especially in the case of business issues that initially ‘disguise’ themselves as IT-only issues, this is of the utmost importance.

These observations will serve as input for the further development/improvement of GEA, while also providing relevant starting points for more quantitative follow up research/evaluations.

7 Conclusion

In this paper, we discussed a real world case study in Business/IT alignment at the strategic level. The specific business issues addressed in the case were: (1) What are the necessary change initiatives of the introduction of this new system? (2) What are the best choices in terms of solution direction and approach? The *coherence dashboard* as configured for the Ministry of SAE, illustrated that Business/IT alignment is not only a matter of aligning “the business” and “the IT” aspects of an enterprise. The SAE case indicates that a more refined perspective is called for, in which multiple aspects need to be aligned with the goal of achieving more coherence.

As discussed in the introduction, in the case of the Dutch Ministry of SAE, the GEA method was a given. However, as also indicated, the GEA method is continuously developed further using a design science rhythm. The lessons learned as listed in the previous Section, have already lead to further improvements of the GEA method. In our further research we will, continue to conduct real life case studies, and based on the findings, further elaborate and improve GEA.

References

1. Balogun, J., Hope Hailey, V., Johnson, G., Scholes, K.: Exploring Strategic Change, 2nd edn. Financial Times Prent.Int (2003) ISBN-13: 9780273683278
2. Greefhorst, D., Proper, H.A.: Architecture Principles – The Cornerstones of Enterprise Architecture. Enterprise Engineering Series. Springer, Berlin (2011), ISBN-13: 9783642202780, http://www.springer.com/business+%26+management/business+information+systems/book/978-3-642-20278-0
3. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28, 75–106 (2004)
4. Simons, R.: Levers of Control: How Managers Use Control Systems to Drive Strategic Renewal. Harvard Business School Press (1994) ISBN-13: 9780875845593
5. Wagter, R.: Sturen op samenhang op basis van GEA – Permanent en event driven. Van Haren Publishing, Zaltbommel (2009) (in Dutch) ISBN-13: 9789087534066
6. Wagter, R., Nijkamp, G., Proper, H.A.: Overview 1th Phase - General Enterprise Architecturing. White Paper GEA-1, Ordina, Utrecht, The Netherlands (2007) (in Dutch)
7. Yin, R.K.: Case Study Research – Design and Methods, 4th edn. Sage Publications (2009) ISBN-13: 9781412960991

Agile Development and UX Design: Towards Understanding Work Cultures to Support Integration

Jennifer Ferreira

University of Calgary, Calgary AB T2N 1N4, Canada
jen.ferreira@ucalgary.ca

Abstract. Organisations are investing heavily into enterprise system infrastructure. With a move to agility and Agile software development, there is an increasing need for understanding how Agile developers and User Experience (UX) designers work together in practice. This paper outlines the current approaches to investigating the combination of Agile development and UX design and indicates a direction for future research that could benefit integration across the cross-functional teams required for enterprise software development.

Keywords: Agile development, User Experience design, integration, studies of work practice, work group culture.

1 Introduction

As enterprise environments embrace agility and Agile software development, there is also a need for understanding how User Experience (UX) designers[1] fit into Agile projects. The benefits of Agile development combined with UX design are well documented in the literature. For practitioners, the advantages include an improved product [1,2], better quality of the user experience [3], increased team confidence [4], team morale [5,6], devotion and satisfaction [7]. For academic researchers, the combination of Agile development and UX design is of interest as part of a broader, on-going effort for better integration of Human-Computer Interaction (HCI) into software engineering [8,9,10].

There are, however, non-trivial problems that Agile development and UX design practitioners have to work through in order to achieve these benefits. There are problems reported regarding the timing and scheduling of Agile development and UX design tasks. Such issues include involving users [11,12] and incorporating their feedback back into the development effort [13,14], coordinating between the activities of the UX designers, Agile developers and other non-Agile teams [11,15] and performing usability testing in the context of other Agile development tests, e.g. unit testing and acceptance testing [16]. Further,

[1] *User Experience (UX) design* refers to a collection of approaches that design the users' experience with the software by setting out to understand users and how they will use the software, and iteratively refining the design.

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 608–615, 2012.

UX designers working with Agile development teams find that time for up-front UX design and user research is short [8,11,16,17,18], and in some cases there is no allocated time for up-front research [3].

Both Agile development and UX design aim to build quality software, but despite their common concern, each approaches development from a different perspective. While Agile methods mainly describe activities addressing code creation (e.g. [19]), UX design methods describe activities for designing the product's interaction with a user [20]. There is little guidance about integrating these two perspectives, and still few detailed accounts providing a close scrutiny of Agile development and UX design being combined in practice. While the growing body of literature continues to focus on process descriptions and recommended techniques, the day-to-day work involved and the many and varied settings in which the techniques and processes are applied, remain largely unexplored.

2 Processes, Techniques and Situatedness

2.1 Processes and Techniques that Establish and Maintain Focus

The combination of UX design with Agile development helps practitioners maintain focus on important aspects of software development. While Agile development focuses on creating working software, UX design focuses on creating a usable design that may or may not be in the form of working software. A central concern is to support the weaknesses of one with the strengths of the other. That is, Agile development is seen to lack an awareness of usability issues, with little guidance for how and when designers contribute to the process (e.g. [15]). According to Patton [4], eXtreme Programming (XP) allowed the team to deliver on time, however, they were still not satisfying the end user. Only by adding the techniques of Usage-Centered Design could they focus on user goals and focus on what was required to build. UX design approaches lack a structured approach to transforming designs into working software and, therefore, little guidance on how developers are involved. Besides a focus on end-users practitioner accounts show a variety of ways in which UX design supports Agile development, such as supporting planning and prioritisation [21], bridging communications between stakeholders [22]. Wilcox et al. [23] also point out the bridging effect of design: "Design, in this and many more cases, acted as a middle-man between the high-level vision and bottom-up innovations under development." Therefore, the two are seen as complementary approaches that, used together, improve the outcome of the software development effort. Examples in the literature combine established Agile approaches, e.g., XP, or Scrum, with established design approaches, e.g., Usage-Centered Design [24], Usability Engineering [25]. We also find examples of well-known HCI techniques such as personas [26] and scenarios [27] being used on Agile projects.

2.2 Dealing with a Piecemeal Approach to Development

Agile values discourage up-front planning activities for software design, i.e., up-front design of the code [28], which often affects time allocated for UX design on Agile projects. When Agile projects allocate little time for performing UX design before implementation begins, and allow requirements to change during development, the holistic view of the product is compromised and in some instances seen to be missing [3,12,13]. It is understood that UX designers require enough time at the outset of the project to perform user research and sketch out a coherent design. Practitioners have been found to do some UX design up front in order to get the development effort started [29,9] and in fact consider up-front UX design more acceptable than up-front code design [30]. Advice is generally to have designers remain ahead of the developers, so that they have enough time to design for what is coming ahead and evaluate what has already been implemented — beginning with a stage before implementation begins, i.e,. "Cycle Zero" [31,3] or "Sprint Zero" [32], then interleaving the activities of UX design work with implementation work throughout the duration of the project (e.g. [31]). To fit with the time-boxed Agile iterations, usability techniques are often adapted to fit within shorter timescales. Lee and McCrickard [13] suggest performing lightweight usability testing as part of the acceptance testing process. This approach is successful when usability tests are smaller, more focused and performed more often. Sy [3] suggests adjusting the timing and granularity of user research.

2.3 The Situated Nature of Practice

Another strand of research moves away from a focus on process descriptions and specific techniques and investigates the combination of Agile development and UX design by engaging with the settings in which practitioners work. These studies are still rare and, as a result, assumptions about practice and the factors that shape how developers and designers work together are largely unexplored. The few existing studies that have taken account of organisational settings have observed practitioners working in everyday contexts. These include the framework for integrating Agile development and user-centred design proposed by Chamberlain et al. [8], the discovery of the importance of identity and vision in successful UX design by Kollmann et al. [33] and the forms of collaboration between developers and designers by Brown et al. [34].

Such studies are an important contribution to understanding how Agile development and UX design practitioners work together, as there is evidence that both developers [35] and designers [36] deviate from prescribed processes in practice. Whether prescribed processes are followed depends on the relation between the individuals and the environment in which they work — which can be referred to as *situatedness*. Suchman introduces the term *situated action* to emphasise that 'every course of action depends in essential ways upon its material and social circumstances' [37, p. 50]. Considering Agile development and UX design work as situated action has the implication that process descriptions, on their own, are not adequate for understanding practice [38]. Complexity in real work settings arises from different balances of power and different levels of influence the designers and developers may have on determining how they work. Therefore, a focus on the day-to-day negotiations of order that take into consideration the wider setting in which the practitioners are embedded allows a better understanding of work arrangements, dependencies and mechanisms that make the integration and coordination of work possible. Further, rather than considering the Agile teams as self-contained units separated from their wider organisational setting, we have to take into consideration the settings in which they carry out their work, i.e. we have to take into account that they may be reliant on other parties for getting their work done and that their wider organisational setting has consequences for how they get their work done.

3 Towards Understanding Work Group Cultures

While there is advice in the literature for how Agile developers and UX designers can mesh their activities (e.g. [31,3]), empirical evidence based on observations of what Agile developers and UX designers actually do suggests that integration relies on recurring efforts in engaging other individuals for their input [39]. This does not follow any of the suggested process models in the current literature. Instead, progress requires the involvement of those with decision-making power, or expertise, when the work demands it. This could include engaging with developers, designers or other members of the organization, depending on how decision-making power is distributed, or the type of expertise required to perform the particular task at hand.

Previous research on contextual factors influencing the integration of Agile development with UX design has emphasised the importance of values and assumptions underlying decision-making [40]. This approach assumes that underlying the decisions concerning how Agile development and UX design should work together in the organisation, are values and assumptions that motivate those decisions — a value being a belief about what ought to be [41, p. 25]. This relationship between decisions, values and assumptions has been studied by other researchers as levels of organisational culture, the visible aspects of which are observable in its artefacts [42]. While the research that gave rise to the importance of values and assumptions was not aimed at understanding organisational culture, elements that can be associated with culture showed up in the analysis of the data and were significant in explaining how and why the Agile developers and UX designers in each team were working together in the ways that had been observed. Explicating values and assumptions gave insights into differing views on how "best" to create quality software and how "best" to create software combining the skills of UX designers and Agile developers.

Agile developers and UX designers do what is required in order to get their job done, by maintaining focus and coordinating between their tasks, expecting certain behaviours from others, maintaining mutual awareness, negotiating progress and engaging with each other. Their work in each setting depends on the values endorsed by the organisations in which the developers and designers are embedded. Two views emerged: (1) the best way of developing quality software is by keeping the Agile developers and UX designers separate, and (2) the best way of developing quality software is via the Agile developers and UX designers working closely together. Those teams embedded in settings endorsing similar values, also had similar experiences in carrying out their work. Further details appear in Ferreira et al. [40]. The significance of articulating these values and assumptions allows them to be discussed and compared, so (in)forming part of a wider, ongoing debate about how UX design and Agile development should be combined. Second, the analysis shows that the integration of UX design and Agile development in practice can not be characterised solely by adaptations of different processes and techniques. Rather, the empirically-based results demonstrate clearly that the nature of UX/Agile practice is characterised strongly in terms of the values of the organisations in which they are embedded. Therefore, instead of merely prescribing processes and adding techniques, improving practice requires also a shift in focus to the explication of contextual values.

One way of taking this discussion further, would be to analyse practice in terms of *culture*. Agile software development research has addressed culture on a global level (e.g. [43]) and at the organisational level (e.g. [44]). Cultures endorse and constrain the behaviours of individuals embedded in organisations [45] and may therefore provide a useful analytical lens for studying Agile development and UX design in practice. Vaughan [46] has shown that culture does not necessarily have to extend to an entire organisation. By studying work group culture the focus can be the "set of solutions produced by a group of people to meet specific problems posed by the situation that they face in common" [46, p. 64]. As the success of enterprise systems rely on the integration of various organisational entities [47], a sympathetic understanding of the cultures endorsed by each entity could aid an understanding of how to better support their interactions.

4 Concluding Remarks

This paper indicates a direction for future research that could benefit integration across the cross-functional teams required for enterprise software development — that of studying work group culture. Current research on the combination of Agile development and UX design places an emphasis on processes and tools, while research addressing practice in the settings in which it unfolds remains scarce. A move towards understanding work group cultures endorsed by the entities involved in enterprise software development could aid an understanding of how to better support their interactions, beyond prescribing processes and techniques that are rarely followed in practice.

References

1. Meszaros, G., Aston, J.: Adding usability testing to an agile project. In: Proceedings of the Conference on AGILE 2006, pp. 289–294. IEEE Computer Society, Washington, DC (2006)
2. Singh, M.: U-SCRUM: An Agile methodology for promoting usability. In: Agile 2008: Proceedings of the Agile 2008 Conference, Toronto, Ontario, Canada, August 4-8, pp. 555–560. IEEE Computer Society, Los Alamitos (2008)

3. Sy, D.: Adapting usability investigations for Agile user-centered design. Journal of Usability Studies 2(3), 112–132 (2007)
4. Patton, J.: Designing Requirements: Incorporating Usage-Centered Design into an Agile SW Development Process. In: Wells, D., Williams, L. (eds.) XP 2002. LNCS, vol. 2418, pp. 1–12. Springer, Heidelberg (2002)
5. Frank, A., Hartel, C.: Feature teams collaboratively building products from ready to done. In: Agile Conference, AGILE 2009, pp. 320–325 (August 2009)
6. Hussain, Z., Lechner, M., Milchrahm, H., Shahzad, S., Slany, W., Umgeher, M., Wolkerstorfer, P.: Agile User-Centered Design Applied to a Mobile Multimedia Streaming Application. In: Holzinger, A. (ed.) USAB 2008. LNCS, vol. 5298, pp. 313–330. Springer, Heidelberg (2008), doi:10.1007/978-3-540-89350-9_22
7. Lindström, H., Malmsten, M.: User-centred design and agile development: Rebuilding the swedish national union catalogue. The Code{4}Lib Journal 5, 12–15 (2008)
8. Chamberlain, S., Sharp, H., Maiden, N.A.M.: Towards a Framework for Integrating Agile Development and User-Centred Design. In: Abrahamsson, P., Marchesi, M., Succi, G. (eds.) XP 2006. LNCS, vol. 4044, pp. 143–153. Springer, Heidelberg (2006)
9. Fox, D., Sillito, J., Maurer, F.: Agile methods and user-centered design: How these two methodologies are being successfully integrated in industry. In: Agile 2008: Proceedings of the Agile 2008 Conference, Toronto, Ontario, Canada, August 4-8, pp. 63–72. IEEE Computer Society, Los Alamitos (2008)
10. Memmel, T., Gundelsweiler, F., Reiterer, H.: Agile human-centered software engineering. In: HCI 2007: Proceedings of 21st BCS HCI Group Conference, University of Lancaster, UK, pp. 167–175. British Computer Society (September 2007)
11. Detweiler, M.: Managing UCD within Agile projects. Interactions 14(3), 40–42 (2007)
12. Wolkerstorfer, P., Tscheligi, M., Sefelin, R., Milchrahm, H., Hussain, Z., Lechner, M., Shahzad, S.: Probing an Agile usability process. In: CHI 2008: CHI 2008 Extended Abstracts on Human Factors in Computing Systems, pp. 2151–2158. ACM, New York (2008)
13. Lee, J.C., McCrickard, S.: Towards extreme(ly) usable software: Exploring tensions between usability and Agile software development. In: Agile 2007: Proceedings of the Agile 2007 Conference, Washington, DC, USA, August 13-17, pp. 59–71. IEEE Computer Society, Los Alamitos (2007)
14. McInerney, P., Maurer, F.: UCD in Agile projects: Dream team or odd couple? Interactions 12(6), 19–23 (2005)
15. Budwig, M., Jeong, S., Kelkar, K.: When user experience met agile: A case study. In: Proceedings of the 27th International Conference Extended Abstracts on Human Factors in Computing Systems, CHI 2009, pp. 3075–3084. ACM, New York (2009)
16. Sy, D., Miller, L.: Optimizing Agile user-centered design. In: CHI 2008: CHI 2008 Extended Abstracts on Human Factors in Computing Systems, pp. 3897–3900. ACM, New York (2008)
17. Ambler, S.W.: Tailoring usability into Agile software development projects. In: Law, E., Hvannberg, E., Cockton, G. (eds.) Maturing Usability. Human-Computer Interaction Series, pp. 75–95. Springer, London (2008)
18. Ungar, J., White, J.: Agile user centered design: Enter the design studio - a case study. In: CHI 2008: CHI 2008 Extended Abstracts on Human Factors in Computing Systems, pp. 2167–2178. ACM, New York (2008)
19. Beck, K., Andres, C.: Extreme Programming Explained: Embrace Change, 2nd edn. Addison-Wesley, Boston (2004)

20. Sharp, H., Rogers, Y., Preece, J.: Interaction Design: Beyond Human-Computer Interaction, 2nd edn. John Wiley & Sons (2007)
21. Patton, J.: Finding the forest in the trees. In: Companion to the 20th Annual ACM SIGPLAN Conference on Object-oriented Programming, Systems, Languages, and Applications, OOPSLA 2005, pp. 266–274. ACM, New York (2005)
22. Broschinsky, D., Baker, L.: Using persona with XP at LANDesk Software, an Avocent company. In: Agile 2008: Proceedings of the Agile 2008 Conference, Toronto, Ontario, Canada, August 4-8, pp. 543–548. IEEE Computer Society, Los Alamitos (2008)
23. Wilcox, E., Nusser, S., Schoudt, J., Cerruti, J., Badenes, H.: Agile Development Meets Strategic Design in the Enterprise. In: Concas, G., Damiani, E., Scotto, M., Succi, G. (eds.) XP 2007. LNCS, vol. 4536, pp. 208–212. Springer, Heidelberg (2007)
24. Patton, J.: Hitting the target: Adding interaction design to Agile software development. In: OOPSLA 2002: OOPSLA 2002 Practitioners Reports, pp. 1–7. ACM, New York (2002)
25. Kane, D.: Finding a place for discount usability engineering in Agile development: Throwing down the gauntlet. In: ADC 2003: Proceedings of the 2003 Agile Development Conference, Salt Lake City, UT, USA, June 25-28, pp. 40–46. IEEE Computer Society, Los Alamitos (2003)
26. Haikara, J.: Usability in Agile Software Development: Extending the Interaction Design Process with Personas Approach. In: Concas, G., Damiani, E., Scotto, M., Succi, G. (eds.) XP 2007. LNCS, vol. 4536, pp. 153–156. Springer, Heidelberg (2007)
27. Obendorf, H., Finck, M.: Scenario-based usability engineering techniques in Agile development processes. In: CHI 2008: CHI 2008 Extended Abstracts on Human Factors in Computing Systems, pp. 2159–2166. ACM, New York (2008)
28. Fowler, M.: Is Design Dead? (2004) published online, http://martinfowler.com/articles/designDead.html#id20933 (updated May 2004) (last accessed June 2011)
29. Ferreira, J., Aumann, Y., Biddle, R.: Up-Front Interaction Design in Agile Development. In: Concas, G., Damiani, E., Scotto, M., Succi, G. (eds.) XP 2007. LNCS, vol. 4536, pp. 9–16. Springer, Heidelberg (2007)
30. Ferreira, J., Noble, J., Biddle, R.: Agile development iterations and ui design. In: Agile 2007: Proceedings of the Agile 2007 Conference, Washington, DC, USA, August 13-17, pp. 50–58. IEEE Computer Society, Los Alamitos (2007)
31. Miller, L.: Case study of customer input for a successful product. In: Agile 2005: Proceedings of the 2005 Agile Conference, Denver, CO, USA, July 24-29, pp. 225–234. IEEE Computer Society, Los Alamitos (2005)
32. Najafi, M., Toyoshiba, L.: Two case studies of user experience design and Agile development. In: Agile 2008: Proceedings of the Agile 2008 Conference, Toronto, Ontario, Canada, August 4-8. IEEE Computer Society, Los Alamitos (2008)
33. Kollmann, J., Sharp, H., Blandford, A.: The importance of identity and vision to user experience designers on agile projects. In: Proceedings of the 2009 AGILE Conference, Chicago, IL, USA, August 24-28, pp. 11–18. IEEE Computer Society (2009)
34. Brown, J., Lindgaard, G., Biddle, R.: Collaborative events and shared artefacts: Agile interaction designers and developers working toward common aims. In: AGILE Conference (AGILE), pp. 87–96 (August 2011)
35. Fitzgerald, B.: The use of systems development methodologies in practice: A field study. Info. Systems J. 7, 201–212 (1997)

36. Gasson, S.: The reality of user-centered design. J. End User Comput. 11(4), 5–15 (1999)
37. Suchman, L.A.: Plans and Situated Actions: The problem of human-machine communication. Cambridge University Press, New York (1987)
38. Ferreira, J., Sharp, H., Robinson, H.: User experience design and agile development: Managing cooperation through articulation work. Software: Practice and Experience 41(9), 963–974 (2011)
39. Ferreira, J., Sharp, H., Robinson, H.: Agile development and user experience design integration as an on-going achievement in practice. Accepted to Agile 2012 to be held in Dallas, TX (2012)
40. Ferreira, J., Sharp, H., Robinson, H.: Values and Assumptions Shaping Agile Development and User Experience Design in Practice. In: Sillitti, A., Martin, A., Wang, X., Whitworth, E. (eds.) XP 2010. LNBIP, vol. 48, pp. 178–183. Springer, Heidelberg (2010), doi:10.1007/978-3-642-13054-0_15
41. Schein, E.H.: Organizational Culture and Leadership, 4th edn. Jossey-Bass, San Francisco (2010)
42. Schein, E.H.: Organizational culture. American Psychologist 45(2), 109–119 (1990)
43. Hossain, E., Bannerman, P.L., Jeffery, R.: Towards an understanding of tailoring scrum in global software development: a multi-case study. In: Proceedings of the 2011 International Conference on Software and Systems Process, ICSSP 2011, pp. 110–119. ACM, New York (2011)
44. Strode, D., Huff, S., Tretiakov, A.: The impact of organizational culture on agile method use. In: 42nd Hawaii International Conference on System Sciences, HICSS 2009, pp. 1–9 (January 2009)
45. Vaughan, D.: Signals and interpretive work: The role of culture in a theory of practical action. In: Cerulo, K.A. (ed.) Culture in Mind: Toward a Sociology of Culture and Cognition, pp. 28–54. Routledge, New York (2002)
46. Vaughan, D.: The Challenger Launch Decision: Risky technology, culture and deviance at NASA. The University of Chicago Press, Chicago and London (1996)
47. Nah, F.F.H., Lau, J.L.S., Kuang, J.: Critical factors for successful implementation of enterprise systems. Business Process Management Journal 7(3), 285–296 (2001)

Viewing Enterprise Resource Planning Systems as Services: A Conceptual View, Based on Practical Experiences, of Designing Information Systems as Services

Nicklas Holmberg and Björn Johansson

Department of Informatics, Lund School of Economics and Management, Lund University, Ole Römers väg 6, SE-22363 Lund, Sweden
{Nicklas.Holmberg,Bjorn.Johansson}@ics.lu.se

Abstract. Viewing information systems (IS) as services is beneficial but still an unexplored approach for IS in organizations. The aim of this exercise is to contribute to the knowledge base on designing IS as services. This paper presents an analysis of enterprise resource planning (ERPs) systems through the lens of service oriented architecture (SOA). Services are explained and defined through SOA theory. IS are explained and defined through ERP theory. This paper contributes to the debate on viewing IS as services by presenting a view of ERPs facilitating to fulfill business needs. This paper is influenced by systems and design thinking, and service oriented IS development. Framed by shared promises between SOA and ERP systems we discuss the question whether SOA or ERP fulfills business needs? The analysis of ERPs from a SOA perspective provides us with the conclusion that the question is not about SOA or ERP but rather to provide SOA architected ERPs. It can be said that by viewing ERPs as services it is clear that the combination of ERPs and SOA could be seen as one way forward when designing software aiming at bridging the gaps supporting IS and business processes and allowing the business shaping the IS.

Keywords: Enterprise Resource Planning, Service Oriented Architecture, Business Processes, Business Rules.

1 Introduction

Viewing Information Systems (IS) as services is beneficial but still an unexplored approach for IS in organizations [1]. Thinking of systems as services enables new systems design methods to emerge [1]. Then, new IS Development (ISD) methods aim to improve business communication and provide practical routes toward increased relevance of IS in business and society [1].

This paper is influenced from practical experiences of a national research project named VacSam. VacSam is a set of composed digital services shaping an Immunization Information System (IIS). VacSam provides unique vaccination recommendations to any foreign child entering Sweden with a purpose to decrease child deaths due to preventable infectious diseases.

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 616–623, 2012.

In the 1990's there were a big hype among organizations to implement standardized software packages i.e., Enterprise Resource Planning (ERPs) systems. Implementation of ERP systems could be seen as the prize organizations had to pay to compete in a constant emerging market. Despite the fact that a service dominant economy emerged and influenced organizations to be recognized as goods or service dominant, not much was done by dominant providers to design ISs as services [1, 2]. Predominant revenues were discovered in e.g., IBM Global Services Business, a department of IBM that did not exist prior to the 1990s [2]. ERP systems must reflect "reality" because they have profound influence on business logic and the way business runs. Similar directions are discussed by Hirschheim and Klein [3] and Taylor and Raden [4]. Business owners have limited influence on ERPs design, thus vendor specific standardized software packages emerging as embedded business actors [5, 6]. It can also be said that implemented ERPs do not fulfill the promises that were indicated by vendors making organizations searching for other solutions. One solution presented is Service Oriented Architecture (SOA), and according to Forrester Research SOA penetration is stronger than ever [7].

From glancing at SOA it can be said that it promises to service orient a business by bridging the gaps between business and process permitting business to shapes IS [8, 9]. From a quick overview of the promises of ERPs it is indicated that also ERPs promise to deliver that solution. However, if ERPs aim at bridging the gap by being service oriented is not clear. That brought us to explore ERPs from a service perspective, - viewing ERPs as services. SOA is used as a lens providing a service with properties and the suggested view with a concrete ground for explanation of what SOA services are. Based on that SOA shares promises expressed by ERPs we question whether SOA or ERP fulfills business needs? The view of ERPs and SOA as separate but related entities is more carefully discussed in future sections of this paper organized accordingly: First we present and define SOA and the concept of services in SOA. The section thereafter defines ERPs and discusses problematic issues with ERP implementation. The reason for doing this is to be able to provide an exploration of designing ERPs as services, which is done in Section 4. In the final section concluding thoughts on what it means to design ERPs as services as well as giving some directions for future studies in this area is presented.

2 Service Oriented Architecture (SOA)

The approach to SOA that we present here departs from a none-technical point of view: 1) SOA as a conceptual architecture, 2) SOA manifesto and the basic principles of SOA and, 3) SOA realizing technologies. The purpose is to decrease the risk of putting SOA on a par with e.g., Web-services, one of many SOA realizing technologies [9]. SOA is a conceptual architecture functioning independent from choice of realizing technology [8]. During the last decade SOA received criticism as an ambiguous buzzword only realizing obsolete application platforms e.g., standardized software packages. In 2007 Gartner [10] predicted less than 25 percent of large companies to manage their SOA projects by 2010. This paper therefore argues that only realizing obsolete application platforms is not the intention of SOA [8, 9, 11]. Just as different designers have different understandings of different material SOA means different "things" depending on whom you ask [12].

Sincere efforts to operationalize SOA have been made. In 2005, Erl [9] established the basic principles for SOA. Eight basic principles could now intrinsically express Separation of Concerns (SoC) and properties for a SOA service. A few years later in 2009, Arsanjani et al., [8] established the SOA Manifesto. Fourteen guiding principles stressed the importance of maintaining a business perspective in any SOA realization [8]. To consider shared services therefore became more important than specific purpose implementations. In 2009, the SOA manifesto, an extended abstract level of SOA, was hierarchically placed above the basic principles of SOA expressing high level business modeling guidelines e.g., 'to respect the social and power structure of the organization' [8]. To achieve architecture supporting the SOA manifesto the basic principles for SOA became of profound importance. The eight principles express properties which a SOA service must possess to be recognized as eligible and responsible. Supporting SoC, the basic principles express modularization, encapsulation and information hiding, also, commonly known in Object Oriented Programming (OOP).

Conceptual, a property of SOA dates back to the origin of "service". That term is the reason to the intrinsic confusion of what SOA is. In the 1930's, the U.S. Department of Commerce's Standard Industrial Classification (SIC) provided a service a code of classification. In the late 1970's Hill [13] provided "service" a definition [2]: "[...] a service is a change in the condition of a person, or a good belonging to some economic entity, brought about as the result of the activity of some other economic entity, with the approval of the first person or economic entity." [13].

Thus, a SOA service changes a condition of a Service Provider (SP), because of an activity, corresponding to a request made by a Service Requestor (SR) to a SP through a transport medium e.g., Internet, with the approval of an SP. Arsanjani et al., [8] suggests that service-orientation, encapsulating service, frames what "one does". Service-orientation of SOA is then interaction between a SR, requesting a service from a SP, providing a service from a Service Directory (SD). That is similar to how Gustiené [14] stressed the importance of interaction as the base for service orientation which must support principles of SoC. Then, "[...] Service-oriented architecture (SOA) is a type of architecture that results from applying service orientation." [8]. While, interaction is "[...] Related mutual actions occurring within a shared space of time or place." [15]. Interaction occurs through a transport medium and its direction is no simplistic association but guideposts indicating orientation of interaction in "reality". Then, an SD-listed-service permits peer to peer communication between SR and SD with approval of SP. Thus, service orientation based on interaction permits a service to become a unit of communication enabling a SR, a SP and a SD, to interact within a shared space on a share time in a known real world direction i.e., SOA.

OASIS and Open Group created formal definitions of SOA with intentions to facilitate SOA's implicit terminology and reduce its different meanings: "A paradigm for organizing and utilizing distributed capabilities […]" [16]. According to the SOA manifesto [8], SOA is realized with varying technologies and standards. Then, SOA functions independent from choice of realizing technology. Based on this chapter we define SOA accordingly: SOA is a paradigm that shapes a conceptual architecture, functioning independent from choice of realizing technology, providing abilities to describe a service, its properties and its orientation, for conscious change or design of a service-oriented business.

2.1 SOA Services

Addressing SOA services addresses SOA realizing technology per see. SOA realizing technology shapes a SOA service as a unit of communication why SOA realizing technology permits interaction. Services responsible for functional components shaping a Business Information Systems (BIS) could thus be viewed as components equipped with logical boundaries forming composable subject matters. Hence, a service could be and, is responsible for the logic it encapsulates independently existing, as an entity of its own right, from other services and ISs.

The laconic brief of SOA realizing technologies are: e.g., Simple Object Access Protocol (SOAP), Universal Description, Discovery and Integration (UDDI), Web Service Description Language (WSDL) etc. Such technologies are architectural styles or patterns solving reoccurring known design problems quite contrary to conceptual SOA [9, 12]. Based on that, there is a plethora of SOA realizing technologies putting the basic principles of SOA into use and supporting SoC. Then the basic principles of SOA are what shape SOA services representing a part of the physical form of a SOA, suggested by Erl [9]: 1) Services are reusable, 2) Services share a formal contract, 3) Services are loosely coupled, 4) Services abstract underlying logic, 5) Services are composable, 6) Services are autonomous, 7) Services are stateless, 8) Services are discoverable [9]. Based on the same conditions we argue that functional areas shaping components of ERPs can be designed as services.

3 Enterprise Resource Planning Systems

The ERP concept is broad and the market of ERP is dominated by a number of few companies including SAP, Oracle, and Microsoft. However, there are a number of key characteristics that more or less all ERP systems have making them a unique subtype of IS: 1) ERPs are standardized packaged software [17] designed with the aim of integrating an entire organization [18-20]. 2) The ERP ought to cover all information processing needs and to integrate the internal value chain with an organization's external value chain through Business Process (BP) integration [18] and 3) Provide the entire organization with common master data [21]. Even if ERPs have a high impact on organization's BPs it is either not used or is implemented in the wrong way [22].

The main problem presented is the misfit between ERP functionality and business requirements. Soh, Kien and Tay-Yap [23] describe this as a common problem when adopting software packages. The problem of "misfit" means that e.g., "Many people feel that the current ERP system has taken (or been given) a role that hinders or does not support the business processes to the extent desire" [6]. Then, ERPs are process-based or at least attempt to be process-based. According to Koch [24] the basic architecture building on a department/stab model as for instance SAP'R/3 makes ERPs not supporting the idea of BPs and thereby not the integration between different departments in an organization. It does not help that the ERP vendor attached some words about BPs onto their ERP if the basic architecture does not support BPs [24].

3.1 Functional Areas of ERP Systems Architecture

ERPs are often described from a functional perspective meaning that the systems architecture mimics a functional organizational description. That implies that each department has its own ERP component. However, the basic architecture of an ERP follows the master data thoughts [21]. Then, functional ERP areas use a unified database. Different ERP vendors describes this in different ways, however, the most common description is to discuss *modules*. Thus, the implementing organization implements a core module and then selects what modules to implement on top of the core module(s). The ERP architecture therefore builds on a vertical organizational description. That implies horizontal work tasks involving different departments which are not clearly described in ERP architecture. Then users of ERPs could understand the ERPs as not supporting the business process they work with, resulting in a misfit between ERP and users interpretation of how the system fulfill their needs.

4 Designing ERP Systems as Services

ERPs as described above, builds to a high extent on functional areas: 1) Inventory, 2) Production, 3) Accounting, 4) HR, 5) Delivery, 6) BI, 7) Sales, 8) Engineering, 9) Production Planning and 10) Purchase. However, the volatile nature of organizations makes it complex to implement the same ERP in all organizations.

Based on the basic principles of SOA, functional areas of an ERP system could be designed as independent components, separated by logical boundaries, designed with the same accuracy as a single class or entity is [9, 25]. That view is based on modularization realized with information hiding and to learn ISD by "doing".

From the description of ERPs it can be stated that it is hard to see if its promises have been fulfilled. The same can be said about SOA promises. However, it seems that if combing the ideas of SOA when designing an ERP that may be a way forward to fulfill promises from both ERP and SOA. Then, the desired result, bridging the gap between BPs and IS so that business shapes IS into what could be described as a SOA architected ERP. The question is then how can SOA improve the design of ERPs? A tentative answer to that question could be that the focus moves from a functional view to a conceptual holistic view, meaning that functions in the ERP, if designed as services, could be seen and provided as applications that could be used in different BPs. In practice this could imply that an organization is permitted to deal with the problem of organizational support with a horizontal supportive IS.

On those conditions, functional areas of an ERP could form components shaped by services eligible to execute in a SOA. Based on practical experiences from VacSam, it is shown that by composing digital services a SOA architected IS can be shaped. However, VacSam is not strictly an ERP system. On the other hand, from this perspective, VacSam corresponds to a component shaped by about 60 digital services used by GPs for diagnosis. Logically, diagnosis is similar to any Business Rules (BR) governed functional area and could most likely be compared to e.g., accounting. That is the foremost reason to why we consider SOA a conceptual architecture applicable in a plethora of contexts and not a pattern for routine design. Thus, SOA permits to design eligible services and thereby service orienting a business regardless of its character. With profound influence on “how” and “what” business runs, functional areas of any IS must reflect reality to be able to support business and processes as a whole thus, bridging the gap. Therefore it is vital for the purpose, entitling the being, of an ERP to support decision points in a BP permitting or constraining its execution.

From research on ERPs we recognize a lack of transparency regarding logic responsibility. What logic that shapes a functional area of an ERP component, is not clear. According to Morgan, [26], Graham, [27] and Von Halle [28] part of business logic shapes decision logic. The other part of business logic is shaped by BPs i.e., process logic [29]. Logics separation through SoC then has profound influence on foundational e.g., alethic logic, and is crucial for IS and ISD success (Holmberg and Steen, 2011). Tentative results of such SoC is consistent automated business logic [28] -a promise made by Business Rules Approach (BRA) familiar as a support to SOA nowadays [26, 27], but, still an unrecognized approach for native ERP design.

BRs of BRA renounce from expressing “who”, “when”, “where” or “how” a business rule executes [26, 28, 30] or any temporal aspects managed by operational process logic of an IS [29] thus BRs express “what” [30]. BRs either constrain BP activities from executing or permit them to execute attaining a state why BRs triggers BPs [31]. A BR could therefore be viewed as a definition or a delineation of an aspect of a business [26, 27, 32]. Then, BRs govern BPs [33]. And, BRs are recognized as the operational decision logic of an IS. Quite contrary, BPs are recognized as the operational process logic of an IS [29]. With that distinction a business’s “what” i.e, decision logic expressed by BRs and, “how” i.e., process logic expressed by BPs, becomes transparent and manageable as separate but interrelated components shaped by business objects advocating IS and business alignment [29, 31].

Without logics separation, decision logic is scattered with process logic and application specific code in the same object, in plural forming components or modules, commonly known as obsolete legacy IS. That makes it hard to recognize what a functional area of an ERP is and which logic each component shaping a functional area of an ERP is encapsulating. Moreover, that would renounce SoC, SOA and BRA by being solely one track minded [25-27, 33, 34]. ERPs could be seen as quite far from supporting SoC, since it implies to “consume an elephant” rather than trying to break down problems into smaller manageable pieces, similar to objectification or break down of connections. That directs us to the conclusion that it would be beneficial viewing ERPs as services.

5 Conclusion

We have learnt that, ERP forces adopting organization to change its BPs. From this view of ERPs, as services, that would not be the case for adopting organizations. The view would rather force BPs or their process logic to shape composable services forming one part of an ERP expressing “how”, to achieve goals. The other part is shaped by BRs or their decision logic as services expressing “what”, to achieve goals. When the two types of services are composed they can be viewed as a component reflecting a functional area of an ERP providing a desired result similar to those provided by components for e.g., diagnosis or accounting. That would then correspond to a SOA architected ERP.

Viewing ERPs as services explicitly renounce from any "silver bullet approach" but implies to break down problems into smaller pieces, supporting principles of SoC, and to systematically design responsible services, supporting SOA, shaping components reflecting functional areas of an ERP in turn supporting business needs, one at a time. The analysis of ERPs from a SOA perspective provides us with the conclusion that the question is not about SOA or ERP but rather to provide SOA architected ERPs. It can be stated that by viewing ERPs as services it is clear that the combination of ERPs and SOA could be seen as one way forward when developing software that aims at bridging the gaps between supporting IS and business processes. However, additional empirical research e.g., DSR on designing functional areas as components, shaped by SOA services, supporting important business problems, followed by evaluation, would cast a better ground for interesting future research on the suggested perspective of ERPs. To the best of our knowledge, this perspective of ERPs, BRs and BPs, is an important area for IS research providing more knowledge on how business and IS is independent but intrinsically related entities of today.

References

1. Alter, S.: Viewing Systems as Services: A Fresh Approach in the IS Field. Communications of the Association for Information Systems 26(11), 195–224 (2010)
2. Chesbrough, H., Spohrer, J.: A Research Manifesto For Services Science. Communications of the ACM 49(7), 35–40 (2006)
3. Hirschheim, R., Klein, H.K.: Four Paradigms of Information Systems Development. Communications of the ACM 32(10), 1199–1216 (1989)
4. Taylor, J., Raden, N.: Smart (Enough) Systems, How to deliver competitive advantage by automating hidden decisions. Prentice Hall, Pearson Education, USA (2007)
5. Melin, U.: The ERP system as a part of an organization's administrative paradox. In: 11th European Conference on Information Systems. Naples, Italy (2003)
6. Askenäs, L., Westelius, A.: Five roles of an information system: a social constructionist approach to analyzing the use of ERP systems. In: Twenty First International Conference on Information Systems, pp. 426–434. Association for Information Systems, Brisbane (2000)
7. Robinson, J.: SOA still kicking, says Forrester (2009), http://www.information-age.com/channels/development-and-integration/perspectives-and-trends/1053022/soa-still-kicking-says-forrester.thtml [cited November 22, 2011]
8. Arsanjani, A., et al.: The SOA Manifesto (2009), 20100930, http://www.soa-manifesto.org/
9. Erl, T.: Service-oriented architecture: concepts, technology, and design. Prentice Hall PTR, Pearson Education, Upper Saddle River (2005)
10. Gartner. Bad Technical Implementations and Lack of Governance Increase Risks of Failure in SOA Projects. Gartner Newsroom (2007), Press Release 20111021, http://www.gartner.com/it/page.jsp?id=508397
11. Erl, T.: SOA Principles, The Service-Orientation Design Paradigm (2008), 20100304, http://www.soaprinciples.com/p3.php
12. Holley, K., Arsanjani, A.: 100 SOA Questions Asked and Answered. Pearson Education, Inc. (2011)

13. Hill, P.T.: On Goods and Services. The Review of Income and Wealth 23(4), 314–339 (1977)
14. Gustiené, P.: Development of a new service-oriented modelling method for information systems analysis and design. Karlstad University, Karlstad (2010)
15. Alexanderson, P.: Adding Audibility, in Department of Informatics. Lund University, Lund (2007)
16. OASIS Group. Reference Model for Service Oriented Architecture 1.0, 20111006 (2006), `http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf`
17. Wieder, B., et al.: The impact of ERP systems on firm and business process performance. Journal of Enterprise Information Management 19(1), 13–29 (2006)
18. Lengnick-Hall, C.A., Lengnick-Hall, M.L., Abdinnour-Helm, S.: The role of social and intellectual capital in achieving competitive advantage through enterprise resource planning (ERP) systems. Journal of Engineering and Technology Management 21(4), 307–330 (2004)
19. Rolland, C., Prakash, N.: Bridging the Gap Between Organisational Needs and ERP Functionality. Requirements Engineering 5(3), 180–193 (2000)
20. Wier, B., Hunton, J., HassabElnaby, H.R.: Enterprise resource planning systems and non-financial performance incentives: The joint impact on corporate performance. International Journal of Accounting Information Systems 8(3), 165–190 (2007)
21. Hedman, J., Borell, A.: ERP systems impact on organizations. In: Grant, G. (ed.) ERP & Data Warehousing in Organizations: Issues and Challenges, pp. 1–21. Idea Group Publishing, Hershey (2003)
22. Millman, G.J.: What did you get from ERP, and what can you get? Financial Executives International, 15–24 (May 2004)
23. Soh, C., Kien, S.S., Tay-Yap, J.: Cultural fits and misfits: Is ERP a universal solution? Communications of the ACM 43(4), 47–51 (2000)
24. Koch, C.: BPR and ERP: Realising a vision of process with IT. Business Process Management Journal 7(3), 258 (2001)
25. Dijkstra, E.W.: On the role of scientific thought. Selected writings on Computing: A Personal Perspective. Springer-Verlag New York, Inc., New York (1982)
26. Morgan, T.: Business rules and information systems: aligning IT with business goals. Addison-Wesley, Boston (2002)
27. Graham, I.: Business rules management and service oriented architecture: a pattern language. John Wiley, Chichester (2006)
28. Von Halle, B.: Business rules applied: building better systems using the business rules approach. John Wiley & Sons, Inc., Computer Publishing, New York (2001)
29. Holmberg, N., Steen, O.: Business Process and Business Rules Modelling In Concert for e-Service Design and Business Alignment. In: 1st International Conference on Cloud Computing and Services Science (CLOSER). Springer, Noordwijkerhout (2011)
30. Date, C.J.: What not How: the business rules approach to application development. Addison-Wesley, Reading (2000)
31. Bajec, M., Krisper, M.: A methodology and tool support for managing business rules in organisations. Information Systems 30(6), 423–443 (2005)
32. Van Eijndhoven, T., Iacob, M.E., Ponisio, M.L.: Achieving Business Process Flexibility with Business Rules. In: 12th International IEEE Enterprise Distributed Object Computing Conference, pp. 95–104 (2008)
33. BRG. The Business Rules Manifesto Version 2.0 (November 1, 2003), 20110104, `http://www.businessrulesgroup.org/brmanifesto.htm`
34. Ross, R.G.: Principles of the business rule approach. Addison-Wesley, Boston (2003)

Author Index